A Systematic Method for Configuring VLSI Networks of Spiking Neurons

Neftci, E ; Chicca, E ; Indiveri, G ; Douglas, R

Abstract: An increasing number of research groups are developing custom hybrid analog/digital very large scale integration (VLSI) chips and systems that implement hundreds to thousands of spiking neurons with biophysically realistic dynamics, with the intention of emulating brainlike real-world behavior in hardware and robotic systems rather than simply simulating their performance on general-purpose digital computers. Although the electronic engineering aspects of these emulation systems is proceeding well, progress toward the actual emulation of brainlike tasks is restricted by the lack of suitable high-level configuration methods of the kind that have already been developed over many decades for simulations on general-purpose computers. The key difficulty is that the dynamics of the CMOS electronic analogs are determined by transistor biases that do not map simply to the parameter types and values used in typical abstract mathematical models of neurons and their networks. Here we provide a general method for resolving this difficulty. We describe a parameter mapping technique that permits an automatic configuration of VLSI neural networks so that their electronic emulation conforms to a higher-level neuronal simulation. We show that the neurons configured by our method exhibit spike timing statistics and temporal dynamics that are the same as those observed in the software simulated neurons and, in particular, that the key parameters of recurrent VLSI neural networks (e.g., implementing soft winner-take-all) can be precisely tuned. The proposed method permits a seamless integration between software simulations with hardware emulations and intertranslatability between the parameters of abstract neuronal models and their emulation counterparts. Most important, our method offers a route toward a high-level task configuration language for neuromorphic VLSI systems.

DOI: https://doi.org/10.1162/NECO_a_00182

Originally published at:
Neftci, E; Chicca, E; Indiveri, G; Douglas, R (2011). A Systematic Method for Configuring VLSI Networks of Spiking Neurons. Neural Computation, 23(10):2457-2497.
DOI: https://doi.org/10.1162/NECO_a_00182

ARTICLE Communicated by Ralph Etienne-Cummings

A Systematic Method for Configuring VLSI Networks of Spiking Neurons

Emre Neftci
emre@ini.phys.ethz.ch
Elisabetta Chicca
chicca@ini.phys.ethz.ch
Giacomo Indiveri
giacomo@ini.phys.ethz.ch
Rodney Douglas
rjd@ini.phys.ethz.ch
Institute of Neuroinformatics, ETH, and University of Zurich, Zurich 8057, Switzerland

An increasing number of research groups are developing custom hybrid analog/digital very large scale integration (VLSI) chips and systems that implement hundreds to thousands of spiking neurons with biophysically realistic dynamics, with the intention of emulating brainlike real-world behavior in hardware and robotic systems rather than simply simulating their performance on general-purpose digital computers. Although the electronic engineering aspects of these emulation systems is proceeding well, progress toward the actual emulation of brainlike tasks is restricted by the lack of suitable high-level configuration methods of the kind that have already been developed over many decades for simulations on general-purpose computers. The key difficulty is that the dynamics of the CMOS electronic analogs are determined by transistor biases that do not map simply to the parameter types and values used in typical abstract mathematical models of neurons and their networks. Here we provide a general method for resolving this difficulty. We describe a parameter mapping technique that permits an automatic configuration of VLSI neural networks so that their electronic emulation conforms to a higher-level neuronal simulation. We show that the neurons configured by our method exhibit spike timing statistics and temporal dynamics that are the same as those observed in the software simulated neurons and, in particular, that the key parameters of recurrent VLSI neural networks (e.g., implementing soft winner-take-all) can be precisely tuned. The proposed method permits a seamless integration between software simulations with hardware emulations and intertranslatability between the parameters of abstract neuronal models and their emulation counterparts. Most important, our method offers a route toward a high-level task configuration language for neuromorphic VLSI systems.

1 Introduction

Developments during the past three decades of computational neuroscience (Schwartz, 1993) have provided a prodigious collection of tools for the simulation of biophysically realistic neurons and their networks on general-purpose digital computers, including NEURON (Hines & Carnevale, 1997), GENESIS (Bower, Beeman, & Wylde, 1998), and PCSIM (Pecevski, Natschläger, & Schuch, 2008). The majority of these simulation tools are designed to encode detailed mathematical models of neurons into a form appropriate for digital simulation. Finally, these tools rest on numerical methods for the simulation of difference equations.

A second approach to computational neuroscience is concerned with the direct physical emulation of neural computation. The intention of these emulations is to understand and exploit for novel computational technologies the physical principles of brainlike computation rather than to simulate its detailed biophysics on general-purpose digital computers. In this article, we are concerned with an emulation method based on the construction of CMOS VLSI neuromorphic devices and systems (Mead, 1989), which comprise large assemblies of silicon neurons and synapses whose dynamics are very similar to those of their biological counterparts (Mahowald & Douglas, 1991). An increasing number of research groups are developing these custom hybrid analog/digital very large scale integration VLSI chips and multichip systems that implement hundreds to thousands of spiking neurons with biophysically realistic dynamics (Silver, Boahen, Grillner, Kopell, & Olsen, 2007; Schemmel, Fieres, & Meier, 2008; Serrano-Gotarredona et al., 2009), as well as analogs of biological vision (Mead & Mahowald, 1988; Culurciello, Etienne-Cummings, & Boahen, 2003; Lichtsteiner, Posch, & Delbruck, 2008; Posch, Matolin, & Wohlgenannt, 2010) and auditory sensors (Lyon & Mead, 1988; van Schaik & Liu, 2005).

Although the electronic engineering aspects of these emulation systems are proceeding well, progress toward the actual emulation of brainlike tasks is restricted by the lack of suitable high-level configuration methods of the kind that have been developed over many decades for simulations on general-purpose computers. The key difficulty is that the dynamics of the CMOS electronic analogs are determined by transistor biases that do not map directly to the parameter types and values used in typical abstract mathematical models of neurons (e.g., the Hodgkin and Huxley neuron model) and their networks. A further difficulty is that neuromorphic electronic circuits often exploit the subthreshold regime of transistor operation in order to match the biological properties of neurons. The signals of these subthreshold circuits are small and therefore susceptible to noise and fabrication variation. As a consequence of these technical difficulties, neuromorphic engineers spend a large amount of time and effort obtaining desired functionality by tuning the many circuit parameters manually and by configuring ad hoc system solutions on a case-by-case basis.

Clearly, a systematic and automated configuration methodology is urgently required to design and implement large-scale brain-inspired computational systems.

This article provides a general method for resolving this problem. We describe a parameter mapping technique that permits an automatic configuration of the voltage and current biases of CMOS VLSI neural circuits so that their electronic emulation conforms to an abstract digital neuronal simulation. We show that the neurons configured by our method exhibit spike timing statistics and temporal dynamics that are the same as those observed in the software-simulated neurons and, in particular, that the key parameters of recurrent VLSI neural networks can be precisely tuned.

Determining unknown parameters and state variables of physical systems by measurement of a limited number of observables is a challenging problem and has been the focus of several research groups (Brillinger, 1998; Keat, Reinagel, Reid, & Meister, 2001; Paninski, Pillow, & Simoncelli, 2004; Okatan, Wilson, & Brown, 2005; Huys, Ahrens, & Paninski, 2006; Abarbanel, Creveling, Farsian, & Kostuk, 2009). Mapping parameters from silicon neural networks to their equivalent theoretical models is analogous to this problem, and so in principle the parameter estimation methods from these works can be applied. However, for the purpose of configuring hardware neurons, we also require reverse mapping, which should be determined by their equivalent theoretical model. In reverse mapping, the unknown parameters are those of the hardware neurons, and the theoretical model parameters represent the desired target values.

Provided a method to estimate a parameter, a typical solution to the parameter configuration problem is to iteratively search the space of biases until the estimated parameter matches a desired criterion. Unfortunately, this approach requires a new measurement from the hardware system at each step of the iteration. This can be prohibitively slow, especially when each neuron (operating in real time) must be probed separately and can be computationally expensive because large amounts of data (e.g., membrane potential traces) must be analyzed.

Such parameter search methods can be improved with the use of heuristics. For example Russell, Orchard, and Etienne-Cummings (2007) demonstrate a multistage evolutionary algorithm that can tune the parameters of a VLSI neural network until its behavior implements the one of a central pattern generator network. Their approach is similar to a black box model in that it does not require any knowledge of the underlying VLSI circuit. Although this method would also allow configuring any neuromorphic neural network, we propose a different bidirectional mapping approach where the known parameters can be mapped directly to the neural hardware by matching a theoretical neuron model to the VLSI neuron.

This bidirectional mapping approach is based on our ability to derive a suitable electronic model against which to perform parameter estimation. We use the firing rate of the neurons as state variables against which we

fit an abstract neuron model, such as the linear threshold unit (LTU), that represents the instantaneous firing rate of biological neurons (Ermentrout, 1994). This case is different from those in which parameters must be derived for biological networks of spiking neurons. In those cases, obtaining a model that defines suitable parameters may be difficult or even impossible because of the complexity of the underlying phenomena or the lack of adequate experimental data. Fortunately, in our case, the definition of a suitable mathematical model of the hardware neurons and synapses is more straightforward, because the designer has full knowledge of the VLSI system. This circuit model can be more or less detailed depending on the choice of simplifying assumptions for the transistors and analog circuit behaviors and on their expected relationship with standard neural network models. Once a suitable circuit model is chosen, the circuit calibration procedure can then be cast as a standard parameter translation problem. Once the parameter translation has been established, it is possible to determine the bias voltages that set the desired properties of the VLSI neural network (such as synaptic weights, time constants, and refractory periods).

In this article, we apply such parameter translations for automatically configuring VLSI neural networks. The method is general and can be applied to any neuromorphic chip implementing neurons that can be configured to behave as LTUs. Here, we demonstrate its functionality using a specific multineuron neuromorphic VLSI device developed in our institute (Indiveri, Chicca, & Douglas, 2006).

The remainder of the article is organized as follows. In section 2.1, we present neuron models implemented in silicon and describe a method for parameter translation between hardware and theoretical models. In section 2.2 we describe the conditions under which the neuronal models implemented as silicon neurons can be approximated by LTUs. We then describe how to determine the sets of bias voltages for setting desired properties of networks of silicon neurons and apply the procedure to a cooperative-competitive network of silicon neurons using a mean-field approach. In particular, in section 3.1, we show how the methodology proposed can be used to tune the gain of recurrently coupled VLSI neurons. In section 3.2 we show that our method can be used to infer the synaptic time constants of the hardware synapses. In addition, we demonstrate that software-simulated networks of integrate-and-fire (I&F) neurons with matched parameters exhibit comparable temporal dynamics. Finally, in section 3.3, we apply the full methodology to configure a VLSI spiking soft winner-take-all (sWTA) neural network and predict its temporal behavior.

2 Material and Methods

2.1 A Low-Power Integrate-and-Fire Neuron Circuit. Many models of neurons have already been implemented in silicon (Mead, 1989; Mahowald

& Douglas, 1991; van Schaik, 2001; Hynna & Boahen, 2001; Indiveri, 2003; Alvado et al., 2004; Simoni, Cymbalyuk, Sorensen, Calabrese, & DeWeerth, 2004; Schemmel, Meier, & Mueller, 2004; Arthur & Boahen, 2004, 2007; Farquhar & Hasler, 2005; Hynna & Boahen, 2006; Wijekoon & Dudek, 2008; Livi & Indiveri, 2009; Yu & Cauwenberghs, 2009; Rastogi, Garg, & Harris, 2009; Massoud & Horiuchi, 2009; Folowosele, Etienne-Cummings, & Hamilton, 2009). Depending on the complexity of the neuron model, the VLSI neuron may require relatively large areas of silicon. For example, silicon neurons implemented with electronic analogs of voltage-gated channels and with a close analogy of the Hodgkin and Huxley (H&H) formalism require a relatively large area of silicon and are thus usually integrated in relatively small numbers on VLSI chips of practical dimensions (Douglas & Mahowald, 1995; Rasche & Douglas, 2000; Alvado et al., 2004; Yu & Cauwenberghs, 2009). As a consequence, the applications of these types of devices have been confined to specific domains, such as hybrid biological-silicon neuron interaction experiments (Renaud, Tomas, Bornat, Daouzli, & Saïghi, 2007).

A family of simpler spiking neuron models that permits the implementation of large, massively parallel networks in VLSI is the I&F model and the focus of this article. I&F neurons integrate presynaptic input currents and generate a voltage pulse analogous to an action potential when the membrane potential reaches a spiking threshold. Their parameters can be related approximately to the properties of biological neurons. Therefore, in principle, they allow the implementation of neuromorphic systems with biologically meaningful parameterization. However, in practice the electronic and model parameters suffer from the matching problem outlined in section 1. Most VLSI implementations of I&F neuron models are based on the Axon-Hillock circuit originally proposed by Mead (1989). This circuit integrates an incoming current onto a capacitor until a high-gain amplifier switches. The positive feedback produces a voltage spike, and the membrane potential is reset to its initial state. This circuit is extremely compact and has been used in a wide range of neural network chips. However, it does not implement a classical $R - C$ type of leaky I&F model, in which the leak is conductance based. Rather, the leak is often implemented using a constant current sink. As a result, a constant input current charges the capacitor linearly in time, until the spiking threshold is reached, and it is therefore called a constant leakage integrate & fire (CLI&F). We focus our analysis on this class of CLI&F neuron models because it is the foundation of the majority of current silicon neuron implementations. In particular, we use the low-power I&F circuit that was originally proposed in Indiveri et al. (2006) and implemented, for example, in the chips described in Camilleri et al. (2007) and Massoud and Horiuchi (2009). This CLI&F neuron circuit is a silicon neuron model with positive feedback, constant leakage, and refractory period. It has been fully described and characterized in Indiveri et al. (2006). For the scope of this article, it is sufficient to observe that the dynamics governing the

membrane potential V_m below firing threshold obey the following differential equation:

$$C\frac{\mathrm{d}}{\mathrm{d}t}V_m = I(t) - \beta + I_{fb}e^{\frac{\kappa}{U_T}(V_m - V_{th})}, \tag{2.1}$$

where C represents the membrane capacitance, $I(t)$ the neuron's input current, β a passive (constant) leak term, I_{fb} a positive feedback current, V_{th} the neuron's spiking threshold voltage, U_T the thermal voltage, and κ the MOSFET subthreshold slope factor (Mead, 1989). Communication with the device is achieved with the address event representation (AER) protocol, which uses spikes (events) to convey information, in a fashion similar to biological neural systems (Lazzaro, Wawrzynek, Mahowald, Sivilotti, & Gillespie, 1993). When the membrane potential reaches the firing threshold, an AER event is produced and V_m is reset to its resting potential, which is equal to 0 for this circuit. After each spike, the membrane potential is actively clamped to the resting potential for a duration referred to as the refractory period.

2.2 VLSI I&F Parameter Translation Using the Linear Threshold Unit Approximation. We can integrate numerically equation 2.1 and use it in neural network software simulators to implement software networks of spiking neurons that faithfully reproduce the dynamics of the silicon neurons. If we could map the parameters of equation 2.1 directly to the circuit's voltage biases, we would have defined the parameter translation between software and hardware neuron models. However, the relationship between voltage biases and model parameters is nonlinear and includes unknown factors, dependent on the implementation details and on the VLSI fabrication process. Furthermore, this procedure alone would not provide us with useful tools for analyzing the network behavior. Indeed, despite its simplicity, the differential equation 2.1, coupled to the neuron's thresholding nonlinearity, yields a nonlinear system that does not have an explicit analytical solution. To extend our theoretical analysis to network properties, it is necessary to link the silicon neuron circuit to LTU models, model neurons that represent the instantaneous firing rate of biological neurons via a threshold-linear transfer function. This is a very useful model, as the linear relationship between the neuron's input current and its output firing rate has often been observed in biological neurons (Ahmed, Anderson, Douglas, Martin, & Whitteridge, 1998). Although LTUs ignore many of the nonlinear processes that occur at the synaptic level and contain, by definition, no spike timing information, their analytical simplicity and their accuracy in representing the activity of neural networks in a wide variety of cases make them a powerful tool for analyzing neural networks (Yuille & Grzywacz, 1989; Ermentrout, 1994; Ben-Yishai, Lev Bar-Or, & Sompolinsky, 1995).

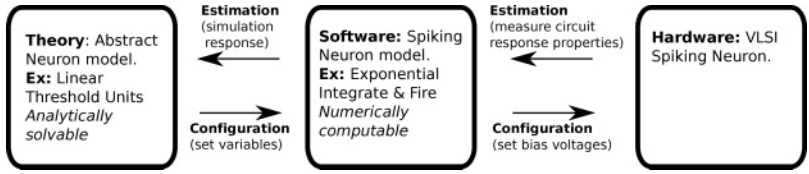


Figure 1: Overview of the parameter translation method for configuring VLSI neurons: How abstract, software, and hardware models of neurons are related and can be used for parameter configuration. The left box represents the set of abstract models that can be analytically solved. The middle box represents the set of spiking neuron models that are typically simulated in software. The right box represents the set of VLSI implementations of spiking neuron models. The left-pointing arrows indicate how parameter estimation is achieved by using observations and measurements from hardware and software models. The right-pointing arrows indicate the parameter configuration process, which originates with the desired biologically relevant variables, and ends with chip biases.

The parameter translation method we propose comprises three components: an abstract analytically tractable neuron model (the LTU), a nonlinear, numerically computable neuronal model (the CLI&F neuron), and a silicon neuron model (such as the VLSI low-power CLI&F neuron). To complete the parameter translation procedure, we must therefore define the mapping between the LTU parameters and the I&F neuron parameters by exploiting their mathematical equivalence and the mapping between the I&F neuron parameters and the voltage biases of the corresponding VLSI circuit. An overview of this procedure is shown in Figure 1.

In the following sections, we show how I&F neurons with dynamics governed by equation 2.1 can be reduced to LTUs and under which constraints. (For a similar study using conductance-based neurons, see Shriki, Hansel, & Sompolinsky, 2003.) The application of this method to other neuron models and networks is possible as long as there exists a regime in which the neurons have a threshold-linear activation function.

2.2.1 From VLSI I&F Neurons to Linear Threshold Units. The positive feedback term of the neuron in equation 2.1 becomes nonnegligible when V_m approaches V_{th}. This term leads to dynamics that are not solvable in the general case. However, if we use an effective firing threshold Θ, the threshold voltage at which the neuron without positive feedback would have fired, producing the same inter-spike interval (ISI) compared to the neuron with positive feedback, we can neglect it. The effective firing threshold Θ can be measured experimentally. In this case, equation 2.1 simplifies to

$$C\frac{\mathrm{d}}{\mathrm{d}t}V_m(t) = -\beta + I(t), \quad V_m(t) \in (0, \Theta), \tag{2.2}$$

where β is a constant leakage current, $I(t)$ the neuron's input current, C the membrane capacitance ($C = 1.06$ pF for this circuit), and Θ the neuron's effective firing threshold (measured $\Theta = 1.1$ V for the neuron threshold voltage $V_{th} = 0.75$ V). Figure 2b compares the membrane potential simulated using the exact dynamics from equation 2.1 with the one simulated using the approximated dynamics given above, stimulated with the synaptic current shown in Figure 2a. In Figure 2c we see that the firing rates of these two models are nearly identical for a large range of input mean frequencies. Equation 2.2 is also the equation that characterizes the Axon-Hillock neuron circuit (Mead, 1989) and many other silicon neurons proposed in the literature. When the membrane potential reaches the firing threshold Θ, the neuron emits a spike of finite pulse width. We can ignore the effects related to the spike pulse widths and the neuron's refractory periods by assuming that the neuron's ISI are much larger than the timescale of such effects. The equation for V_m has an analytical solution that can be determined by integrating equation 2.2. In appendix A, we show that a CLI&F neuron injected with a constant current I_{inj} fires at rate ν,

$$\nu = \frac{1}{C\Theta} \max(I_{inj} - \beta, 0), \tag{2.3}$$

where the $\max(\cdot, 0)$ is a rectification nonlinearity that keeps the firing rate positive.

Input currents are generally provided by other neurons through synapses and are time varying. If the time constant of the soma is small compared to the synaptic time constant, then the firing rate of the neuron tightly follows the synaptic input. Under this condition, equation 2.3 is also a good approximation for time-varying synaptic inputs, and the temporal evolution of the system is governed by the synaptic dynamics (Dayan & Abbott, 2001). This condition is not a limiting factor for the vast majority of neural hardware implementations because the synaptic time constants can usually be set by the user and because this regime must be achieved for the parameter translation calibration step only (explained in detail in section 2.4). Synaptic currents depend on the activities of the presynaptic neurons and are commonly modeled using first-order linear filter dynamics (Destexhe, Mainen, & Sejnowski, 1998). In this model, the output current of a synapse, I_{syn}, in response to an arbitrary spike train $\rho(t)$, is governed by the following equation,

$$I_{syn}(t) = \frac{q_w}{\tau} \cdot e^{-\frac{t}{\tau}} \int_0^t \mathrm{d}s \, e^{\frac{s}{\tau}} \rho(s), \tag{2.4}$$

where q_w is a scalar representing the weight of the synapse (the amount by which I_{syn} is incremented at the arrival of a spike), τ its time constant, and $\rho(t)$ the presynaptic spike train, modeled by a sum of delta Dirac functions:

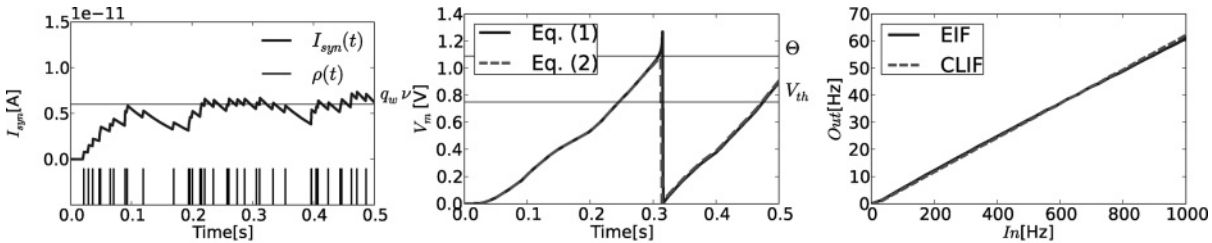


Figure 2: Software simulations comparing the CLI&F model with the exponential feedback term, equation 2.1, to the approximated model in equation 2.2. (a) An example of a software simulated synaptic current, I_{syn}, in response to a Poisson spike train ρ of constant mean frequency (vertical black lines). Because I_{syn} has a long synaptic time constant ($\tau = 100\,\text{ms}$) compared to the input ISI ($\nu = 75\,\text{Hz}$, ISI $\cong 13.3\,\text{ms}$), I_{syn} fluctuates around a steady-state value $q_w\nu$ (see text). (b) A comparison of the membrane potential traces of a software simulated neuron model following equation 2.1 to the approximated model that follows equation 2.2, where Θ is the effective firing threshold. Both neurons are injected with the same synaptic current I_{syn} shown in panel a. Although a slight discrepancy between the two models is observable during the spike generation, in panel c, we see that the firing rates of these two models are nevertheless identical for a large range of input mean frequencies and that the input-output relationship is threshold linear. Panel c also illustrates that the LTU model can capture the mean firing rates of these two neurons models fairly accurately. Values used for the simulation: $C = 1\,\text{pF}$, $\Theta = 1.09\,\text{V}$, $V_{th} = 0.75\,\text{V}$, $U_T = 25\,\text{mV}$, $\beta = 1 \cdot 10^{-12}\,\text{pA}$, $I_{fb} = 5 \cdot 10^{-16}\,\text{pA}$, $\kappa_p = 0.66$, $q_w = 0.08 \cdot 10^{-11}\,\text{C}$, $\tau = 100\,\text{ms}$.

$\rho(t) = \sum_k \delta(t - t_k)$. Synaptic circuits that reproduce such dynamics can be implemented in analog VLSI using a differential-pair integrator (DPI) (Bartolozzi & Indiveri, 2007). To cast the network dynamics in terms of LTUs, we must relate the presynaptic neuron's firing rate to the synaptic current at the postsynaptic neuron. For this, in appendix A, we show that the synaptic current I_{syn} is a low-pass filtered version of the presynaptic neuron's firing rate $\nu(t)$, and that if the synaptic time constant τ is large compared to the input inter-spike interval ISI, then $I_{syn}(t)$ obeys the following differential equation:

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} I_{syn}(t) + I_{syn}(t) \cong q_w \nu(t). \tag{2.5}$$

This equation underlines the fact that under the assumption on the synaptic time constant, I_{syn} integrates the spike responses and fluctuates around a value $q_w \nu$. Figure 2a shows an example trace of I_{syn} (solid line) when the synapse is stimulated with a Poisson spike train $\rho(t)$ of constant mean frequency (vertical black lines) and illustrates how I_{syn} fluctuates around its steady-state value (horizontal line) as a result of the stochasticity of the spike train ρ. Recalling equation 2.3, we can now express the firing rate of the postsynaptic neuron i as a function of both its synaptic currents and a constant injection current I_{inj},

$$\nu_i(t) = \frac{1}{C\Theta} \max\left(I_{inj_i} + \sum_j I_{syn_{ij}}(t) - \beta, 0 \right), \tag{2.6}$$

where the sum runs over all the indexes of presynaptic neurons contacting neuron i. Equation 2.6 depends on synaptic currents $I_{syn_{ij}}$ defined by equation 2.5, which in turn depend on the firing rates of the presynaptic neurons. Therefore, both equations are required to describe the network's firing rate dynamics. The variable $\nu_i(t)$ is the output of a LTU and faithfully models the mean firing rate of a CLI&F neuron model. Figure 2c shows the threshold-linear behaviors of the firing rate of the software-simulated neuron from equation 2.1 and of the approximated CLI&F from equation 2.2 (without the positive feedback term) when stimulated with Poisson spike trains.

2.2.2 Mapping Linear Threshold Unit Variables to VLSI I&F Neuron Parameters. By mapping the CLI&F model variables to the VLSI I&F neuron parameters, we can establish a direct link between analytically tractable LTUs and silicon neurons. Using equations 2.5 and 2.6, we find that the steady-state mean firing rate of a silicon neuron is

$$\nu_i = \max\left(\sum_k w_{ik} \nu_k + b_i - T_i, 0 \right), \tag{2.7}$$

where the variables w_{ij}, T_i, and b_i are defined as

$$w_{ik} = \frac{q_{w_{ik}}}{C\Theta}, \quad T_i = \frac{\beta_i}{C\Theta}, \quad b_i = \frac{I_{inj_i}}{C\Theta}, \tag{2.8}$$

and the terms $w_{ik}\nu_k$ represent the mean input current produced by the synapses, the terms T_i represent a constant leakage, and b_i the experimentally applied input currents. The I_{inj_i} and β_i variables in the translation equations correspond to the neuron's constant injection and the leak currents, respectively, and C and Θ represent the neuron's membrane capacitance and effective firing threshold, respectively. In the following section, we use the parameter translations to configure the mean activity and gain in cooperative and competitive network of VLSI I&F neurons.

2.3 Parameter Configuration in Cooperative and Competitive Networks of I&F Neurons. Cortical neural networks are characterized by a large degree of recurrent excitatory connectivity and local inhibitory connections. This type of connectivity among neurons is remarkably similar across all areas in the cortex (Douglas & Martin, 2004). It has been argued that a good candidate model for a canonical microcircuit, potentially used as a general-purpose cortical computational unit in the cortices, is the soft winner-take-all (sWTA) circuit (Douglas & Martin, 2004), or the more general class of cooperative and competitive network (CCNs) (Amari & Arbib, 1977). A CCN is a set of interacting neurons in which cooperation is achieved by local recurrent excitatory connections, and competition is achieved by a group of inhibitory neurons, driven by the excitatory neurons and inhibiting them (see Figure 3). As a result, CCNs perform both common linear operations and complex nonlinear operations. The linear operations include analog gain (linear amplification of the feedforward input, mediated by the recurrent excitation or common mode input) and locus invariance (Hansel & Sompolinsky, 1998). The nonlinear operations include nonlinear selection or sWTA behavior (Amari & Arbib, 1977; Dayan & Abbott, 2001; Hahnloser, Sarpeshkar, Mahowald, Douglas, & Seung, 2000), signal restoration (Dayan & Abbott, 2001; Douglas, Mahowald, & Martin, 1994), and multistability (Amari & Arbib, 1977, Hahnloser et al., 2000).

We will apply the parameter translation method described in this article to a VLSI device implementing a CCN low-power I&F neurons with DPI synapses (Indiveri et al., 2006, Bartolozzi, Mitra, & Indiveri, 2006). The chip has been fabricated using a standard AMS 0.35 μm CMOS process, and covers an area of about 10 mm^2. It contains 124 excitatory neurons with local hard-wired self; first, second, and third nearest-neighbor recurrent excitatory connections; and 4 inhibitory neurons (all-to-all bidirectionally connected to the excitatory neurons). Each neuron receives input currents from a row of 32 afferent plastic synapses that use address

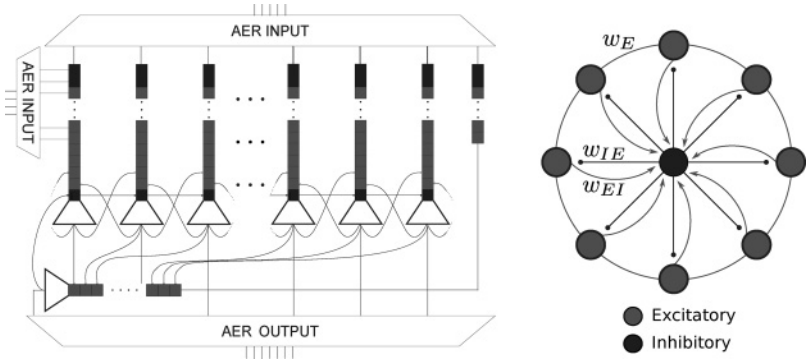


Figure 3: CCN of VLSI neurons with recurrent excitatory couplings and global inhibition. (Left) Circuit diagram of the VLSI chip implementing a spiking CCN. In the circuit, there are 124 excitatory neurons (tiled horizontally) and 4 inhibitory neurons (only one is drawn here to avoid clutter). The dark and light boxes represent inhibitory and excitatory synapses, respectively and the trapezoids represent the somata of the neurons. The excitatory neurons are coupled to each other in a local nearest-neighbor fashion through the synapse of weight w_E. In our chip, the first, second, and third neighbors are coupled to each other. To avoid clutter, only the first nearest-neighbor connections are shown here. A subset of the synapses can be stimulated through the AER (AER input blocks). The spiking activities of all the neurons are encoded as address events (AER output block). (Right) Schematic illustration of the CCN architecture implemented on the chip. When the excitatory neurons become active, the inhibitory neurons receive excitatory input (through a synapse of weight w_{EI}). When the inhibitory neuron becomes active, it inhibits the excitatory neurons back (through a synapse of weight w_{IE}). A network with such connectivity can perform soft winner-take-all computation (see text).

event representation (AER) to receive spikes (Lazzaro et al., 1993). The spiking activity of the neurons is also encoded using the AER. In this representation, input and output spikes are real-time asynchronous digital events that carry analog information in their temporal structure. We can interface the chip to a workstation for prototyping experiments using dedicated boards (Chicca et al., 2007; Fasnacht, Whatley, & Indiveri, 2008). These boards allow us to stimulate the synapses on the chip (e.g., with synthetic trains of spikes) and monitor the activity of the I&F neurons.

2.3.1 Cooperative and Competitive Network with Uniform Input. The LTU approximation presented in section 2.2 can be applied to the CCN of VLSI neurons. If the neural activity is statistically independent from neuron

to neuron, one can use a mean-field approach to study network activity. Such statistical independence typically arises in the diffusion approximation (Tuckwell, 1988), which is accurate when the synaptic weights q_w are small relative to the firing threshold Θ, the number of afferents to each neuron is large, and the network spiking activity is asynchronous (Brunel, 2000; Fusi & Mattia, 1999). Although the diffusion approximation is exact only for infinite-size networks, it is also known to be a good approximation for finite-size networks (Renart, Brunel, & Wang, 2003). The assumptions for the diffusion approximation can be approximatively valid in the case of the VLSI CCN because each excitatory neuron is connected to its first, second, and third neighbors; the synaptic weights can be set to arbitrarily low values; and the inhibitory couplings can be tuned such that the network activity is asynchronous (Brunel, 2000). When the CCN is stimulated with a uniform input and the recurrent excitatory weight w_E is weak enough such that the sWTA does not break the symmetry of the network (e.g., by selecting a winner as a result of small fluctuations), then the CCN can be studied as a function of two LTUs—one for the excitatory population (ν_E) and one for the inhibitory population (ν_I). Under the assumptions stated above, a straightforward calculation (carried out in section A.2) shows that the steady-state activity of the excitatory neurons of the CCN is

$$\nu_E = \frac{b_E}{\Lambda} + \frac{T_I N_I w_{IE} - T_E}{\Lambda}, \tag{2.9}$$

$$\Lambda = 1 - 2w_E + N_I N_E w_{IE} w_{EI},$$

where w_E is the weight of the local recurrent nearest-neighbor excitation (the factor 2 is due to the number of neighbors for each excitatory neuron), w_{EI} is the weight of the excitatory synapse on the inhibitory neuron, w_{IE} is the weight of the inhibitory synapse on the excitatory neurons, T_E and T_I are the thresholds of the excitatory neurons and the inhibitory neurons, respectively, N_E is the number of excitatory neurons, and N_I is the number of inhibitory neurons. The two terms in equation 2.9 represent the activation due to the constant current injection (first term) and the threshold of the inhibitory and excitatory neurons (second term). The system gain of the CCN is Λ^{-1}. Here, for simplicity, we have considered only the first nearest-neighbor couplings. Using equation 2.8, the excitatory population activity ν_E given by equation 2.9 can be cast in terms of currents I_{inj}, β and synaptic weights q_w. The synaptic weight of the differential-pair integrator synapse q_w itself is a function of three currents and a spike pulse duration (see appendix B for a description of these currents). In practice, such currents are controlled by bias voltages in a nonlinear fashion and must be determined at least once experimentally during a calibration procedure, as we shall see in the following section.

2.4 Parameter Translation Calibration. In the previous section, we saw the mathematical link between the LTU and the CLI&F neuron and argued that the latter is a good model for the VLSI I&F neuron. To calibrate the parameter translation, we need to match the VLSI I&F neuron to its hardware instantiation in the chip. This calibration procedure must be carried out only once and corresponds to the arrows pointing to the left in Figure 1. It is equivalent to measuring the unknown parameters that depend on the fabrication process by using the spiking activity data of the chip. To carry out this calibration step, we make use of the steady-state solution given in equation 2.9.

2.4.1 Current-Voltage Characteristics of MOSFET Transistors in Subthreshold Regime. The currents appearing in the translation variables and the synaptic weights in equation 2.8 involve constants related to the properties of the transistors, which in turn depend on their fabrication process and must be measured independently. We describe how this can be done using only spiking activity measurements.

The current-voltage relationship of the MOSFET transistors operating in subthreshold and in saturation have the following expressions (Mead, 1989):

$$I(V_g) = I_{0_n} \frac{W}{L} e^{\frac{\kappa_n}{U_T} V_g} \qquad \text{(n-FET, source node tied to ground)}, \quad (2.10)$$

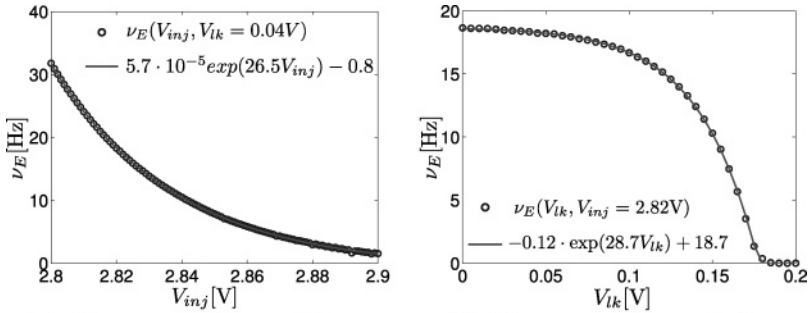
$$I(V_g) = I_{0_p} \frac{W}{L} e^{\frac{\kappa_p}{U_T}(V_{dd} - V_g)} \qquad \text{(p-FET, source node tied to } V_{dd}\text{)}, \quad (2.11)$$

where I_{0_n} and I_{0_p} are the leakage currents (also called off-currents) and κ_n and κ_p are the subthreshold slope factors of the n-FET and the p-FET transistors, respectively. $\frac{W}{L}$ is the width-to-length ratio of the transistor, V_g is the gate voltage, and U_T is the thermal voltage ($\cong 25.6\,\text{mV}$ at 25°C). These relationships are valid only when the source node of the n-FET (p-FET) transistor is tied to ground (V_{dd}), which is the case for all the transistors associated with the currents appearing in equation 2.8.

Measurement of I_{0_p}, κ_p: The current injection is varied with the bias V_{inj} and the leak is kept constant, with all recurrent synapses turned off ($w_{AER}, w_E, w_{EI} = 0$). We fit the firing rate of the excitatory population ν_E to equation 2.9 with the expressions for the transistor currents in equation 2.11:

$$\nu_E(V_{inj}) = b_E(V_{inj}) - T_E \quad \text{(p-FET measurement)}.$$

Measurement of I_{0_n}, κ_n: We perform a similar experiment as the one above except that the leak bias is varied and the current injection



(a) Mean population activity as a function of current injection bias V_{inj}

(b) Mean population activity as a function of leak bias V_{lk}

Figure 4: Two experiments to estimate the off-currents and subthreshold slope factors. (a) The population activity of the excitatory neurons ν_E, averaged over 2 s, as a function of the injection current bias V_{inj}. Since the injection involves a p-FET, the firing rate increases as V_{inj} is set below $V_{dd} = 3.3\,\text{V}$. (b) ν_E as a function of the leak bias V_{lk}. At $V_{lk} > 0.18V$, the VLSI neurons are no longer able to overcome the threshold, and the activity remains at 0.

bias V_{inj} is kept constant. We measure the firing rate of the excitatory neuron population and fit it with

$$\nu_E(V_{leak}) = b_E - T_E(V_{leak}) \quad \text{(n-FET measurement)}.$$

The measurements of ν_E, averaged over all the excitatory neurons, and their corresponding fits are shown in Figure 4 and the fitted values of I_0 and κ are

$$\begin{aligned} I_{0\,n} &= 5.6 \cdot 10^{-14}\,\text{A} \qquad \kappa_n = 0.76, \\ I_{0\,p} &= 4.0 \cdot 10^{-16}\,\text{A} \qquad \kappa_p = 0.69. \end{aligned} \tag{2.12}$$

The two previous measurements determine the values of I_0 and κ for the n-FET and the p-FET. In theory, provided that the size of every transistor is measured, all the variables required for equation 2.9 can be determined. However, this procedure is inaccurate because the large number of synapses per neuron and the various circuits tied to the soma and the synapses often give rise to parasitic currents and also because in our case, the pulse duration, which appears in the expression of the synaptic weight q_w, cannot be precisely estimated. Furthermore, measurements of the I_0 currents are known to be unreliable for small transistors (the typical sizes of the measured transistors were on the order of 300 to 600 λ^2, where λ equals one-half the feature size). The largest contribution to transistor mismatch is due

to doping concentration variations in the silicon and is often described as "spatial white noise" (Pavasović, Andreou, & Westgate, 1994) and therefore follows a gaussian distribution. Although we can assume that the estimates for the I_0's are sufficiently accurate because the two previous experiments were carried out over populations of transistors, the transfer of these measurements to single (small) transistors can lead to imprecise predictions. For these reasons, in the next paragraphs, we individually fit the weight of the couplings required to implement sWTA behavior: (1) the external AER synapse, (2) the excitatory nearest-neighbor synapse, (3) the excitatory synapse of the inhibitory neuron, and (4) the inhibitory synapse of the excitatory neurons, as a function of their biases.

Thus far, the neurons have been stimulated using a constant current injection to the soma, represented by the term b_E in equation 2.9. As mentioned in section 2.3, the neurons can be stimulated externally through an AER synapse (e.g., by means of a digital PC or another chip-producing spike). This type of stimulation is more flexible because the neurons can be individually addressed with time-varying inputs, in contrast to the constant current injection I_{inj} which is controlled by a global bias and is therefore identical for all the neurons. According to equation 2.4 in steady state, a neuron receiving a spike train of constant mean frequency ν_{in} is stimulated with an average input current equal to $w_{AER}\nu_{in}$, where w_{AER} is the weight of the AER input synapse. In this case, the b_E terms in equation 2.9 can simply be replaced or summed with $w_{AER}\nu_{in}$ where necessary:

1. **Mean population activity as a function of external address event representation synapse weight bias $V_{w_{AER}}$.** We measure the firing rate of the excitatory neurons stimulated with spike trains of constant mean frequency as a function of the AER synapse weight bias $V_{w_{AER}}$ with local excitatory and inhibitory couplings turned off (w_E, $w_{EI} = 0$). The measured data are fitted to

$$\nu_E = w_{AER}(V_{w_{AER}})\nu_{in} - T_E, \tag{2.13}$$

   where b is the injection current, ν_{in} is the mean frequency of the input spike trains, w_{AER} is the weight of the synapse, and T_E is the leak (see Figure 5a).

2. **Mean population activity as a function of lateral couplings.** We measure the firing rate of the excitatory population as a function of the recurrent nearest-neighbor synapse bias V_{w_E}. The measured data are fitted to

$$\nu_E = \frac{b_E - T_E}{1 - 2w_E(V_{w_E})}, \tag{2.14}$$

   Results are shown in Figure 5b.

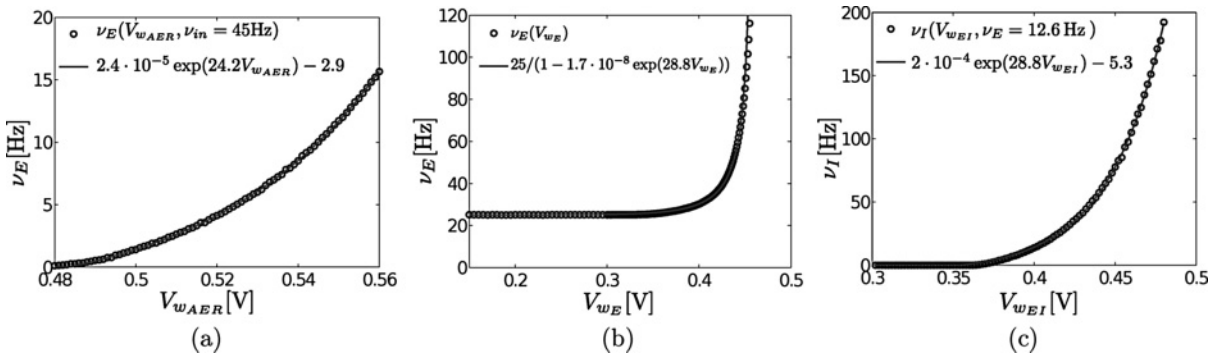


Figure 5: Parameter translation calibration, excitatory synapses. To determine the precise relationships between the synaptic weights and the biases, we perform the calibration described in section 2.4 and present the results of each experiment. (a) The activity of the excitatory neurons, stimulated with spike trains of constant mean frequency against the excitatory AER synapse's weight bias $V_{w_{AER}}$ and fitted to equation 2.13. (b) The activity of excitatory neurons is plotted against the local nearest-neighbor synapse weight bias V_{w_E} and fitted to equation 2.14. (c) The activity of the inhibitory neurons is plotted against the excitatory to inhibitory couplings bias $V_{w_{EI}}$ and is fitted with equation 2.15.

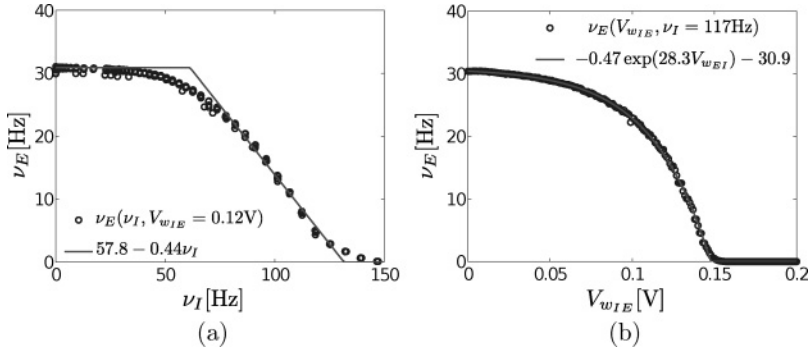


Figure 6: Parameter translation calibration, inhibitory synapses. To determine the precise relationships between the synaptic weights and the biases, we perform the calibration described in section 2.4 and present the results of each experiment. (a) We observe that the effect of the inhibitory synapse is threshold linear with a smooth onset, as described in appendix B. (b) The activity of the excitatory neurons is plotted against the inhibitory synapse weight bias $V_{w_{IE}}$ and fitted to equation 2.16.

3. **Mean population activity as a function of excitatory to inhibitory couplings.** We measure the weight of the excitatory to inhibitory synapse by setting a constant injection b_E to the excitatory neurons and measuring ν_I as a function of $V_{w_{EI}}$. The firing rate of the inhibitory neurons as a function of $V_{w_{EI}}$ is then fitted to

$$\nu_I = N_E w_{EI} \nu_E(V_{w_{EI}}) - T_I, \tag{2.15}$$

where N_E is the number of excitatory neurons and ν_E is kept fixed (results are shown in Figure 5c).

4. **Mean population activity as a function of inhibitory to excitatory couplings.** We measure the inhibitory synaptic current by injecting a constant current to both inhibitory neurons and excitatory neurons and by recording the mean activity of the excitatory neurons as a function of $V_{w_{IE}}$. Due to a nonlinearity in the VLSI implementation of the inhibitory synapse, the inhibitory synaptic current behaves approximately in a threshold-linear fashion (see appendix B). Under these assumptions, the firing rate becomes

$$\nu_E = b_E - T_E - N_I w_{IE}(V_{w_{IE}}) \max(\nu_I - T_I, 0), \tag{2.16}$$

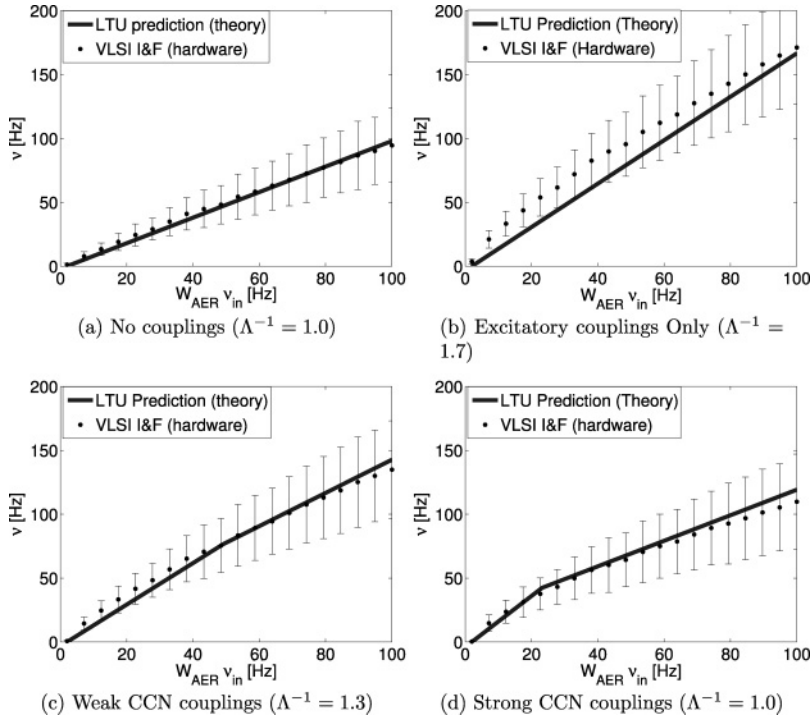
where N_I is the number of inhibitory neurons and T_I the effective threshold due to the nonlinearity. The results are shown in Figures 5c and 6a. The effect of the inhibitory synapse nonlinearity is shown in Figure 6b.

3 Results

3.1 Cooperative and Competitive Network Gain Configuration. The gain of a CCN is an important parameter that characterizes the sWTA operation (Douglas et al., 1994). Experimentally, the transfer function of the CCN is obtained by measuring the steady-state response of the excitatory neurons to spike trains of increasing mean firing rates. By combining the results of the previous section, we can determine every variable on the right-hand side of equation 2.9 as a function of bias voltages, which can be used to calculate the transfer function of the VLSI CCN (see Figure 7). The excitatory and the inhibitory neurons activate sequentially due to the leak of the neurons, resulting in a point where the slope of the transfer function changes abruptly. Up to an input of 50 Hz, only the excitatory neurons are active and show a high gain (steep slope) due to the effect of the excitatory couplings. When both excitatory and inhibitory neurons are active, the gain decreases to Λ^{-1}, as defined in equation 2.9. Except for the discrepancies noticeable in Figure 7b, which are mainly due to nonlinearities in the hardware synapses, the experimental transfer functions (black points) are comparable to those predicted by the LTUs (dark curves) over a large range of configurations, as shown in Figure 7. Some discrepancies are also noticeable at very low and very high firing rates and are mainly due to the nonlinearities in the hardware synapses and the LTU approximation.

3.2 VLSI I&F Neuron Temporal Dynamics Configuration. We have demonstrated that the LTU approximation is accurate for setting the gain of a CCN of VLSI I&F neurons and in predicting its steady-state mean firing rate. However, as many aspects of computation in the brain are achieved during transients, we show that the parameter translation can also predict the temporal dynamics of the VLSI I&F neurons. We first show that the time constants of the DPI synapse are reliably inferred and then perform hardware and software simulations of neurons to compare the step response and spike timing statistics.

3.2.1 Estimating the Postsynaptic Current of the DPI Synapse. The excitatory postsynaptic current (EPSC) of the local nearest-neighbor synapses is the excitatory current flowing into the postsynaptic neurons as a result of presynaptic spiking activity at an excitatory synapse. The EPSC can be determined by recording spike-triggered averages of the membrane potential of an excitatory neuron excited by one of its nearest neighbors, differentiating it and subtracting the leak. To set the time constant of the synapses, we used the measurements of the off-currents and the subthreshold slope factors from section 2.4 and our knowledge of the DPI synapse circuit, recapitulated in appendix B. Because the transistors responsible for controlling the time constant and the weight of the synapse have relatively small sizes ($700\,\lambda^2$ and $140\,\lambda^2$, respectively), we expect a large variance in the EPSCs



(a) No couplings ($\Lambda^{-1} = 1.0$)

(b) Excitatory couplings Only ($\Lambda^{-1} = 1.7$)

(c) Weak CCN couplings ($\Lambda^{-1} = 1.3$)

(d) Strong CCN couplings ($\Lambda^{-1} = 1.0$)

Figure 7: Transfer function of a CCN of 20 excitatory neurons and 4 inhibitory neurons. We measure the activity of 20 VLSI excitatory neurons stimulated by Poisson spike trains of increasing mean frequency (black). We compare the transfer function of the hardware CCN with analytical predictions using equation 2.9 (thick dark line). The stimulation lasted 3 s for each point, and the activity was measured after 2 s, such that the system was guaranteed to reach steady state (the system typically converged to its steady state after 500 ms or less). The curves in panels c and d have two different slopes (gains). This is due to the effect of the leak in the inhibitory neurons: at an input frequency of approximately 50 Hz (c) and 20 Hz (d), the inhibitory neurons start to activate. The transfer functions of the hardware CCN match the LTU solution quite precisely, but less in panel b. This is partly due to the stochastic nature of the stimulation and to nonlinearities in the hardware synapses. For all panels, the weight of the AER input synapse was set to $w_{AER} = 0.5$. The transfer functions were plotted against $w_{AER}\nu_{in}$ in order to emphasize the system's gain.

measured for each individual synapse. However, since the off-currents were measured for the entire populations of neurons, we expect that the EPSC averaged over the entire array will be close to the predictions of equation 2.4.

For these reasons, the measurements of the EPSC are repeated over 120 different local nearest-neighbor synapses of the chip and are compared to equation 2.4 (see Figure 8). To observe the spike-triggered average of the EPSC, the neuron was not allowed to fire. As a result, the membrane voltage was measured in a regime where the leak transistor partially operated out of saturation ($V_m < 0.1\,\mathrm{V}$). This was compensated by modeling the leak transistor in the ohmic region (Mead, 1989). In normal operation, the constant leakage approximation is nevertheless accurate because the leak transistor of the VLSI neuron is in saturation in approximatively 90% of the dynamical range of the membrane potential. The error bars in Figure 8 represent the variation due to the mismatch in the transistors. The predicted EPSC matches the average EPSC accurately, although no direct fit of the EPSC had been performed. We conclude that our parameter translation is accurate for determining the average time constant and average weight of the DPI synapses across the array of neurons.

3.2.2 Comparison with Software-Stimulated CLI&F Neurons. We compare the spike-timing-related statistics of a CCN of VLSI I&F neurons and software-simulated CCN of CLI&F neurons with matched parameters.

For this, we run the following experiment: both software and hardware networks were stimulated with identical Poisson spike trains of constant mean frequency 110 Hz and of 2 s duration. In the software simulations, the effect of transistor mismatch was simulated by sampling the synaptic weights from a gaussian distribution of standard deviation $0.2 \cdot w$ (according to the mismatch observed in the histogram in Figure 8e) and mean w.

Since the particular instance of the synaptic weights in the software simulations does not match the mismatch in the VLSI I&F neurons, we expect the spiking activity to be identical in the statistical sense only. This can be observed in an ISI histogram averaged over the excitatory neurons (see Figure 9c), in which both networks show a nearly identical ISI distribution. The error bars represent the standard deviation due to the distribution of synaptic weights.

The instantaneous firing rates of the software CCN and the hardware CCN are plotted against time in Figure 9d and show that the temporal dynamics of both networks are similar. We conclude that the parameter translation can be used to predict the stationary statistics of the system.

3.3 Soft Winner-Take-All Gain and Time Constant Configuration. The sWTA is a nonlinear function often used in computations requiring decisions, such as saliency detection and object identification.

Equations 2.5 and 2.6 are also appropriate for studying the CCN transients because in the regime $\tau_{syn} \gg$ ISI, the dynamics are dominated by the synapses' time constant (Fourcaud & Brunel, 2002).

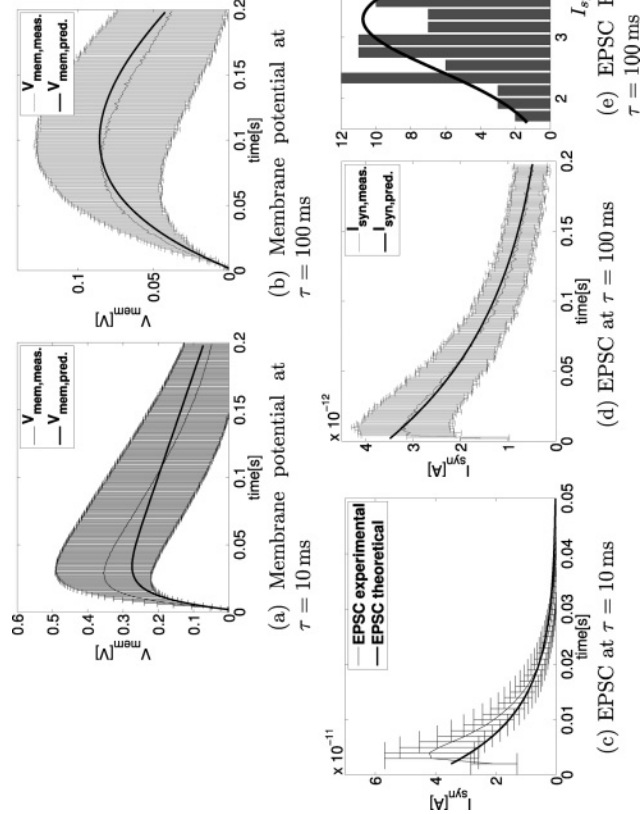
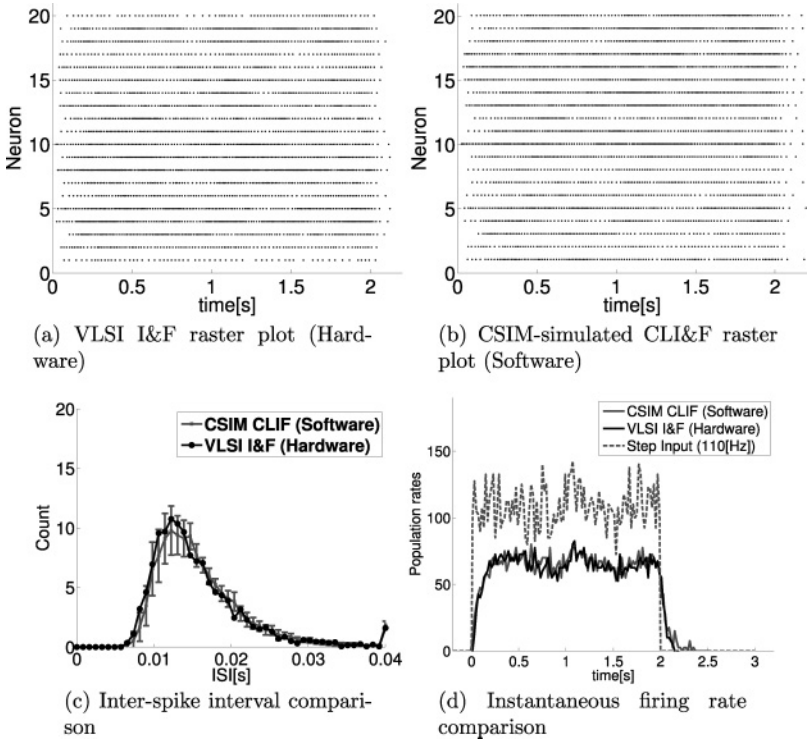


Figure 8: The time constant of the VLSI synapse is inferred by the parameter translation. We measure spike-triggered averages of the membrane potential of a VLSI excitatory neuron excited by its nearest neighbor for time constants (a) $\tau = 10$ ms and (b) $\tau = 100$ ms. The black lines show the excitatory postsynaptic potential predicted by the parameter translations. (c, d) The EPSC is computed by taking the derivative of the membrane potential and adding the leak term. The experimental data for $\tau = 100$ ms matched the predicted curves more accurately because the calibration of the parameter translation had been performed at large synaptic time constants for better accuracy of the LTU approximation. The error bars indicate the standard deviation due to the mismatch in the transistors. (e) A histogram of the EPSC presented in panel d ($\tau = 100$ ms) at its maximum average value (at $t = 0.013$ s) and its gaussian fit with mean $\mu = 3.3 \cdot 10^{-12}$ A and $\sigma^2 = 8.2 \cdot 10^{-13}$ A (black curve).



(a) VLSI I&F raster plot (Hardware)

(b) CSIM-simulated CLI&F raster plot (Software)

(c) Inter-spike interval comparison

(d) Instantaneous firing rate comparison

Figure 9: The spiking activity of the excitatory VLSI I&F neurons statistically matches those of software simulated neurons. We present raster plots of the excitatory neurons from the experiment described in Figure 7 for $w_{AER}\nu_{in} = 65$ Hz. (a) Spiking activity of the VLSI I&F neurons. (b) Spiking activity of software simulated CLI&F neurons. Because of the particular instance of mismatch in the software simulations, the spiking activity matches those of the VLSI neurons in the statistical sense only. This is observed in the ISI distribution (c) and the instantaneous firing rate of the entire population (d). Both hardware and software neurons have equal steady-state firing rates and show comparable responses to a step input (dashed line).

To study the dynamics of the CCN, we carry out an experiment where two excitatory populations are stimulated and compete through the inhibitory population. The mean-field analysis used for the parameter translation calibration in section 2.3 can be extended by adding one LTU per additional excitatory population. The conditions for applying the mean-field approach to the network must then be verified for each additional LTU. This means that the input profile to the neurons of each excitatory LTU must be uniform (but can be different from LTU to LTU). Due to the CCN

connectivity, the two excitatory populations compete through an inhibitory population. When both excitatory populations are stimulated with spike trains of different firing rates, after a short transition period, the activity of the excitatory population receiving the largest input will be amplified (the winner) and the activity of the other population will be suppressed. Because several neurons remain active, the network is said to perform sWTA (as opposed to "hard" sWTA). We set the gain and the time constants of the VLSI CCN using our parameter translation method and compare them with the analytical predictions obtained from the LTU model. In Figure 10c we present the response of the VLSI device in the configuration without local recurrent connectivity and in Figure 10f in the configuration with local recurrent connectivity. The LTU predictions and the activity of the VLSI neurons are comparable, demonstrating that our parameter translation method can be used as a method to efficiently configure the key parameters of an sWTA—its gain, selectivity (ratio between winner and losers activity), and time constant.

4 Discussion

Many research groups are developing custom hybrid analog-digital VLSI chips and systems that implement hundreds to thousands of spiking neurons with biophysically realistic dynamics. However, unlike for conventional digital systems, there exists no high-level programming language to configure them to carry out a desired computation. This is one of the major obstacles to the application of neuromorphic multineuron chips as general-purpose computing devices.

A crucial and necessary step to reach this goal is to determine the transistor biases that map to the parameter types and values used in typical abstract mathematical models of neurons and networks. In this article, we have described a general method for obtaining this mapping in a systematic way.

Our method permits the automatic configuration of VLSI neural networks so that their electronic emulation conforms to a higher-level neuronal simulation. Indeed, we have shown that the neurons configured by our method exhibit spike timing statistics and temporal dynamics that are comparable to those observed in the software-simulated neurons, and in particular that the key parameters of recurrent VLSI neural networks implementing soft winner-take-all can be precisely tuned.

The parameter configuration of VLSI neurons consists in determining the circuit's bias voltages that correspond to the desired model parameters. This problem can be solved by an iterative search that sweeps the various bias voltages on the chip while measuring their effect, until the desired performance is attained. Such global search strategies are generally prohibitive in terms of the time required to set the biases, acquire the experimental data, and analyze them. Furthermore, this brute-force approach offers no

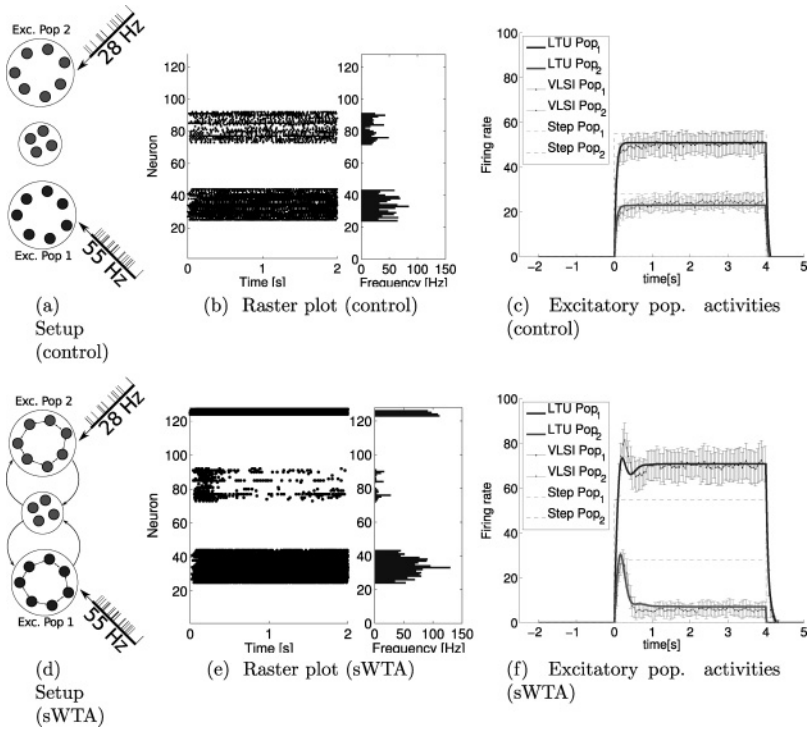


Figure 10: Configuration of key soft winner-take-all parameters. Two excitatory populations, denoted Pop1 and Pop2, are both stimulated by two Poisson spike trains. Pop1 receives a stronger input (55 Hz) and Pop2 a weaker input (28 Hz). (b) The raster plot of both excitatory populations in the case without recurrent couplings. (c) We see that the mean output activities of each populations are equal to the input minus $T_e = 5$ Hz (dashed line). (e, f) The neurons are recurrently coupled (excitation and inhibition); hence, the two populations excite themselves and compete with each other through the inhibitory population (middle population). After a short transition period, the activity of the excitatory population receiving the largest input (the winner) is amplified to 75 Hz and the activity of the losing population is suppressed (to 5 Hz). We see that the response of the hardware neurons is close to the analytical predictions (thick lines). Network parameters were $\Lambda^{-1} = 1.0$, $w_E = 0.3$, $w_{EI} = 0.05$, $w_{IE} = 0.15$, $W_{AER} = 0.5$, $T_E = 5$, $T_I \cong 50$ Hz $\tau_{exc} = 100$ ms, $\tau_{inh} = 100$ ms, $\tau_{AER} = 50$ ms.

predictive power, meaning that the algorithm must be run each time the model parameter is configured to a new value.

On the other extreme, detailed analog circuit simulations using programs such as SPICE can be carried out to determine the biases, with the advantage

that any parameter in the chip can be computed. But often these computed biases are quite different from the ones the chip requires due to inaccuracies in the SPICE models (Tsividis, 1998).

Our parameter translation approach combines the advantages of the two previous solutions by the use of a suitable model of the electronic circuits, against which the parameter estimation is performed. Once the parameter translation is calibrated, the method has sufficient predictive power to tune the parameters of the circuit to any desired value within reasonable accuracy, without any additional measurements.

Compared to a black box approach (Russell et al., 2007), our approach has the advantage that the user can directly map neural models from previous theoretical investigations or software simulations onto the neuromorphic hardware. One drawback of using this approach is the need to build the parameter translations and calibrate them, which requires the detailed knowledge of the circuit that is configured. However, since large-scale implementations usually consist of several stereotypic circuits (e.g., the CCN network), this task is greatly simplified.

Using mean-field theory, we have extended this procedure to networks of neurons. Specifically, we showed that the gain of a CCN implementing the sWTA function could be configured using the LTU to VLSI I&F equivalence. We have also demonstrated that the equivalence of the different models (LTU, CLI&F, VLSI I&F) is accurate in the temporal domain. Indeed, the inferred time constants of the configured hardware synapses were identical to those of the software simulation, and we observed that the ISI's distribution and instantaneous firing rates of the hardware neurons compared well to those of software simulations.

For networks other than the CCN, the application of this method is possible as long as the network can be configured to reach a regime that can be modeled using a mean-field approach. Fortunately, this is the case for most multineuron chips with reconfigurable AER connectivity (e.g., using an AER mapper board; Chicca et al., 2007) and tunable synaptic weights. For configuring chips incorporating few neurons and few possible connections, such that the conditions for applying the mean-field approach cannot be guaranteed, the use of heuristics (Russell et al., 2007) or parameter sweeps is likely to be more efficient.

Many research groups working in the field of neuromorphic engineering continue to use ad-hoc solutions for configuring their systems (see, e.g., Serrano-Gotarredona et al., 2009). While such approaches are viable for implementing networks with few nodes and chips, it will be difficult to scale such systems to implement arbitrary functionality and size.

The large-scale neuromorphic project Neurogrid (Silver et al., 2007), composed of 16 multineuron chips, is aiming to reach a 1-million-neuron AER infrastructure. The current prototype setups used, for example, in modeling orientation selectivity hypercolumns (Choi, Merolla, Arthur, Boahen, & Shi, 2005) and studying synchrony in the gamma band (Arthur

& Boahen, 2007), were configured using an exhaustive search around manually determined operating points.

Another large neuromorphic project has taken place within the Fast Analog Computing with Emergent Transient States (FACETS) project (2005–2009). In this project, Schemmel et al. (2008) have targeted a wafer-scale implementation of spiking neural networks. The goal of this hardware is to speed up simulations typically carried out using digital computers and to perform systematic explorations of a neural network's parameter space. As a consequence, the neurons are designed to operate at about 1000 to 10,000 times faster than their biological counterparts. For configuring their hardware, the FACETS researchers have defined a common interface for simulations and emulations of spiking neurons, which has resulted in the creation of a simulator-independent description language named PyNN (Davison et al., 2008). PyNN, however, does not offer a concrete, general-purpose solution to the parameter configuration problem per se. Instead, its implementation is left to the designer of the hardware interface. In Brüderle (2009), for example, the configuration method uses a brute-force iterative approach, based on comparisons with software simulations performed using PyNN. The parameters are varied until they reach a user-defined target value within a given tolerance. Although the accelerated nature of the FACETS hardware is adequate for such iterative methods, the main bottlenecks in the approach in Schemmel et al. (2008) remain the acquisition and analysis of the AER data.

In our approach, once the parameter translation is calibrated, the desired properties of the neurons can be set (configured) without proceeding through systematic parameter sweeps. Therefore, the model-based approach proposed in this work is a possible solution to dramatically speed up the search for bias voltages, especially in multidimensional parameter search scenarios. In general, due to the interdependence of the biases' effect on the neural output, a parameter search method must search in a space whose dimension is equal to the number of biases controlling the parameter (i.e., $O(N^p)$, where p is the number of biases). Because the initial calibration step measuring the $I0$ and the κ values are carried out in a separate experiment (involving current injection and leak transistors), our method is not affected by such interdependencies. Therefore, the parameter translation method is useful for decreasing the number of measurements required to reach the desired behavior.

Also, both methods can be combined by using the parameter translation result as a starting point in the search method. This operation would still be of order $O(N^p)$ but with a much smaller multiplicative constant. Although we have not carried out a full characterization of the number of measurements required for the search methods because it strongly depends on the parameter that is configured and the optimization method used, we observed that a standard Newton method converged after about 30 measurements when setting I_{inj} (a one-dimensional search task)

compared to about 150 measurements when starting from a reasonable initial condition.

The LTU approximation used in this work holds only when the time constants of the synapses are long in comparison to the ISIs of the neurons. This implies that the calibration step mentioned above—the initial matching of the models—can be performed only in this regime. However, in practice, this is not a limitation since one is free to choose the regime in which the calibration is carried out. The properties of the neurons set outside the LTU regime are still accurate because the mapping between the CLI&F model and the VLSI neuron model remains true (corresponding to the links between the middle and the right boxes in Figure 1). We have demonstrated this by successfully mapping parameters to synapse time constants of 10 ms (see Figure 8b), despite the fact that the LTU approximation is not accurate at such synaptic time constants.

A major problem of analog VLSI devices is the mismatch in their transistor properties, which severely affect the functionality of the computations they carry out. This is especially true when the subthreshold circuits are used to implement sWTA networks, as the recurrent connections amplify the mismatch in the circuits.

Because our method is based on a mean-field approach, the calibration procedure is less sensitive to mismatch in the transistors inherent in the fabrication process. As each LTU represents a population of neurons, we can apply the values estimated during the calibration, the $I0$ currents and the subthreshold slope factor κ, to predict the LTU's behavior.

Nevertheless, when the recurrent couplings in the sWTA become very strong, the network amplifies all the discrepancies due to transistor mismatch, hardware nonlinearities, and the inevitable imprecisions during the calibration, and we observe that the accuracy of the predictions obtained with the parameter translations gradually decreases (see, e.g., Figure 7b).

This imprecision could result in a lack of robustness when implementing high-level computational models that crucially depend on the gain of the underlying recurrent circuits. A possible solution to this problem is to use the proposed parameter translation to provide an initial, coarse operating regime, in which other methods such as on-chip plasticity rules (Mitra, Fusi, & Indiveri, 2006) or homeostasis (Bartolozzi & Indiveri, 2006; Neftci & Indiveri, 2010) can be applied to fine-tune the network's properties.

Our method permits an integration between software simulations with hardware emulations and intertranslatability between the parameters of abstract neuronal models and their emulation counterparts. The key result shown in this article is that it is possible to perform a neural model mapping from theoretical models down to the neuromorphic hardware with a quantitatively accurate parameterization. From this point of view, the parameter translation method raises the neural hardware and the computation they are able to achieve to a level of usability that until now was unaccessible by neuromorphic engineers. This ability can be expected to accelerate research on

hardware emulation of interesting neuronal computational processes, such as hardware applications of liquid state machines (Maass, Natschläger, & Markram, 2002), and the implementation in spiking neurons of graphical models (Steimer, Maass, & Douglas, 2009).

Rutishauser and Douglas (2009) have shown how state-machines can be composed of interconnected networks of winner-take-all (WTA). Since our method can be used to configure the necessary circuits, the way is now open to devise a high-level language that translates a description at the state-machine level down to the biases required to instantiate that functionality at the level of neuromorphic electronic circuits (Neftci, Chicca, Indiveri, Cook, & Douglas, 2010).

Appendix A: Detailed Calculation from CLI&F to LTU Equations

We model each synapse as a first-order low-pass filter (as suggested in Destexhe et al., 1998). In this model, the postsynaptic current in response to a presynaptic spike train $\sum_k \delta(t - t_k)$ (where $\delta(t)$ is the delta Dirac) is given by

$$I_{syn_{ij}}(t) = e^{-\frac{t}{\tau_{ij}}} \frac{q_{w_{ij}}}{\tau_{ij}} \int_0^t \mathrm{d}s \sum_k \delta(t - t_k) e^{\frac{s}{\tau_{ij}}}, \tag{A.1}$$

where $I_{syn_{ij}}$ is the postsynaptic current delivered by the synapse i to the postsynaptic neuron j, τ_{ij} is the time constant of the synapse, and $q_{w_{ij}}$ is a scaling factor for the strength of the synapse.

In other words, for each incoming spike, the synaptic current undergoes a jump of height $\frac{q_{w_{ij}}}{\tau_{ij}}$ and otherwise decays exponentially with a time constant τ_{ij}. By definition of the delta Dirac function, we have

$$I_{syn_{ij}}(t) = \frac{q_{w_{ij}}}{\tau_{ij}} \sum_k U(t - t_k) e^{\frac{t_k - t}{\tau_{ij}}}, \tag{A.2}$$

where $U(t)$ is the unity step function. We will now show how $I_{syn_{ij}}$ relates to the firing rate of the presynaptic neuron. Taking the temporal average around time t, defined by $\langle \cdot \rangle_t = \frac{1}{T} \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} \mathrm{d}s$, we have

$$\langle I_{syn_{ij}}(t) \rangle_t = q_{w_{ij}} \frac{1}{T} \sum_{\{k | t_k \in [t-\frac{T}{2}, t+\frac{T}{2}]\}} 1 - e^{\frac{t_k - t - T}{\tau_{ij}}}.$$

The sum runs over all the spikes occurring between $t - \frac{T}{2}$ and $t + \frac{T}{2}$ and counts the number of spikes $n_T(t)$ that occurred during that time interval. By definition of the firing rate, we have $\frac{n_T(t)}{T} \cong \nu(t)$, and if $t + T$ is large compared to t_k, which is the case for most of the spikes occurring during the interval $[t - \frac{T}{2}, t + \frac{T}{2}]$, then the exponential term can be neglected. Furthermore, if τ_{ij} is much larger than the ISI, then the synaptic current

does not fluctuate too much around its temporal average, and we can assume $\langle I_{syn_{ij}}(t)\rangle_t \cong I_{syn_{ij}}(t)$. Then the synaptic current can be rewritten in the differential form as

$$\tau_{ij}\frac{\mathrm{d}}{\mathrm{d}t}I_{syn_{ij}}(t) + I_{syn_{ij}}(t) \cong q_{w_{ij}}\nu_i(t), \tag{A.3}$$

where the index of ν_i refers to the firing rate of neuron i. Next, we look at how the synaptic current causes the neuron to fire. The input current to a neuron consists in a constant injection I_{inj_i} and a synaptic input $\sum_j I_{syn_{ij}}(t)$. The dynamics of the membrane potential V_{m_i} of neuron i are then given by

$$C\frac{\mathrm{d}}{\mathrm{d}t}V_{m_i}(t) = \sum_j I_{syn_{ij}}(t) + I_{inj_i} - \beta. \tag{A.4}$$

Due to the membrane capacitance C and the leak β, the firing rate is approximately a low-pass filtered version of the synaptic input current. But if this low-pass filter effect is small in comparison to the one occurring at the synapse, then the firing rate of the neuron follows the changes in the synaptic current almost instantly (Dayan & Abbott, 2001). We can assume that this is the case, as the vast majority of neural hardware can be configured with long synaptic currents. This means that we can assume $I_{syn_{ij}} \cong constant$ during one spike generation. By integrating equation A.4 with respect to time, we have

$$C\,V_{m_i}(t) \cong \left(\sum_j I_{syn_{ij}}(t) + I_{inj_i} - \beta\right)t$$

with the initial condition $V_{m_i}(0) = 0$. Writing $\frac{1}{\nu_i}$ as the time required to reach the firing threshold Θ, and solving for ν_i leads to

$$\nu_i(t) \cong \frac{1}{C\Theta}\max\left(\sum_j I_{syn_{ij}}(t) + I_{inj_i} - \beta, 0\right), \tag{A.5}$$

where the term $\max(\cdot, 0)$ comes from the fact that ν cannot be negative. In this final equation, we have shown that CLI&F neurons behave approximately as LTUs that receive their input from other LTUs through linear synapses.

A.1 Self-Consistent Solution for Networks with Uniform Synaptic Dynamics. In the paragraphs above, we argued that the activity of CLI&F neurons is well approximated by equations A.3 and A.5. In this appendix,

we derive a self-consistent set of differential equations describing the state of a network of LTUs. Let us consider a population of N neurons, each of them able to have a synapse with every other neuron and itself.

We can write the synaptic current $I_{syn_{ij}}$ provided by synapse i to the postsynaptic neuron j as a product of $\frac{q_{w_{ij}}}{\tau_{ij}}$ and a dimensionless synaptic variable x_{ij} describing the state of the synapse. By defining the following variables,

$$x_{ij} = \tau_{ij} \frac{I_{syn_{ij}}}{q_{w_{ij}}}, \qquad w_{ij} = \frac{q_{w_{ij}}}{C\Theta}, \qquad T = \frac{\beta}{C\Theta}, \qquad b_i = \frac{I_{inj_i}}{C\Theta},$$

the LTU equations, A.3 and A.5, can be written in a single line for the synaptic variables x_{ij}, noting that at steady state, $x_{ij} = \tau_{ij} \nu_i$. Then the dynamics of N^2 LTUs are given by the following N^2 self-consistent coupled differential equations,

$$\frac{\mathrm{d}}{\mathrm{d}t} x_{ij} + \frac{x_{ij}}{\tau_{ij}} = \max\left(b_{ij} + \sum_k \frac{w_{ik}}{\tau_{ik}} x_{ik} - T, 0 \right) \quad \forall i,\ j{=}1, \ldots, N, \tag{A.6}$$

where the sum over k runs over the indexes of all neurons having synapses with neuron i. We can reduce the number of equations by noticing that all efferent synapses with identical time constants have identical dynamics. In other words, the postsynaptic currents to afferent neurons are identical up to a scaling factor, which is provided by the weight of the synapse q_w. As a result, the state of the synapse can be described by a single synaptic variable per type of synapse per presynaptic neuron. A cartoon of this simplification is shown in Figure 11. In the case of one type of synapse per neuron (i.e., all the time constants of efferent synapses are equal), we get the following N equations, which describe the dynamics of N synaptic variables through N coupled differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t} x_i + \frac{x_i}{\tau_i} = \max\left(b_i + \sum_k \frac{w_{ik}}{\tau_k} x_k - T, 0 \right) \quad \forall i = 1, \ldots, N. \tag{A.7}$$

Once the solution for the synaptic variables $x_i(t) \quad \forall i = 1, ..., N$ is computed, the firing rates $\nu_i(t)$ can be recovered with

$$\nu_i(t) = \max\left(b_i + \sum_k \frac{w_{ik}}{\tau_k} x_k(t) - T, 0 \right), \tag{A.8}$$

Note that according to equation A.3, at steady state, we have $\nu_i = \frac{x_i}{\tau_i}$.

A.2 Closed-Form Solution for the CCN with Uniform Input. The connectivity of the linear threshold unit can be represented by a weight matrix, and in the case of nearest-neighbor connectivity and global inhibition, it is

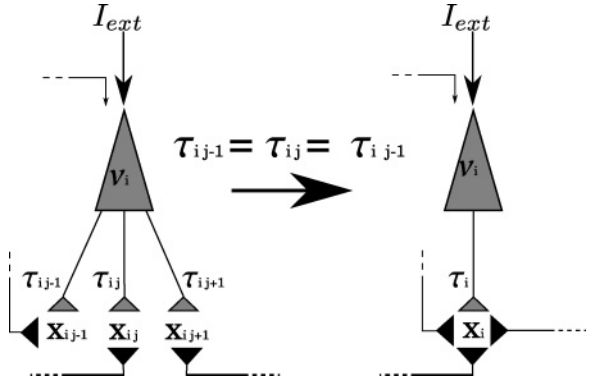


Figure 11: Synapses with identical time constants can be grouped together to simplify LTU network analysis. If $\tau_{ij} = \tau_i \, \forall j$, then the dynamics of the synapses of neuron i can be described by a single synaptic variable x_i. The postsynaptic currents (currents flowing to filled triangles) will differ only by a scaling factor given by the weight of the single synapse.

given by

$$\mathbf{W} = \begin{pmatrix} w_S & w_E & 0 & 0 & w_E & w_{IE} \\ w_E & w_S & w_E & 0 & 0 & w_{IE} \\ \cdots & & \cdots & & & \\ 0 & w_E & w_S & w_E & 0 & w_{IE} \\ 0 & 0 & w_E & w_S & w_E & w_{IE} \\ w_E & 0 & 0 & w_E & w_S & w_{IE} \\ w_{EI} & w_{EI} & w_{EI} & w_{EI} & w_{EI} & 0 \end{pmatrix},$$

where w_E is the connection weight between two nearest neighbors, w_S to the neuron itself, w_{EI} from excitatory neurons to inhibitory neurons, and w_{IE} from inhibitory to excitatory.

If the synaptic weights q_w are small relative to the range $(0, \Theta)$ (where Θ is the effective firing threshold), the number of afferent connections to each neuron is large, and the network activity is asynchronous, a mean-field approach can be used to study the network activity (Brunel, 2000; Fusi & Mattia, 1999). The assumptions above can be valid in the case of the VLSI CCN because each excitatory neuron is connected to six of its neighbors and because the synaptic weights can be set to arbitrarily low values.

If all excitatory neurons receive identical input, $b_i = b \, \forall i$, and if the recurrent couplings are weak, the steady state corresponding to $\{x_k = x_E | k \text{ is excitatory}\}$ and $\{x_k = x_I | k \text{ is inhibitory}\}$ is the only stable solution. This is true if the input can overcome the threshold (leak) of the excitatory

and inhibitory neurons and if the spectral radius of the linearized system (obtained by neglecting the threshold nonlinearity) is below 0 (Hahnloser, Seung, & Slotine, 2000). This is guaranteed if $2w_E + w_s < 1$ (Neftci, Chicca, Indiveri, Slotine, & Douglas, 2008).

In steady state and with uniform input, equation A.7 is written:

$$\begin{aligned}\frac{x_E}{\tau_E} &= \max\left(b + \sum_{j=1}^{N} w_{Ej}\frac{x_j}{\tau_j} - T_E,\ 0\right),\\ \frac{x_I}{\tau_I} &= \max\left(\sum_{j=1}^{N} w_{Ij}\frac{x_j}{\tau_j} - T_I,\ 0\right),\end{aligned}$$

where the indexes E, I, respectively stand for excitatory and inhibitory and w_{ij} are the i j matrix elements of $\mathbf{W}$ and N the total number units. We apply $\nu = \frac{x}{\tau}$, and writing N_E and N_I, the number of excitatory and inhibitory units, respectively, we get:

$$\begin{aligned}\nu_E &= \max(b + 2w_E\nu_E + w_s\nu_E - N_I w_{IE}\nu_I - T_E,\ 0),\\ \nu_I &= \max(N_E w_{EI}\nu_E - T_I,\ 0).\end{aligned} \tag{A.9}$$

Due to the presence of the max(·) term, the firing rates of the excitatory and inhibitory neurons as a function of the current injection b are piecewise linear:

$$\nu_E(b) = \begin{cases} 0 & b \le T_E \\ \dfrac{1}{1-\lambda_E}(b - T_E) & T_E < b < T_I(1-\lambda_E)/(N_E w_{EI}) + T_E, \\ \dfrac{1}{\Lambda}(b + T_I N_I w_{IE} - T_E) & b \ge T_I(1-\lambda_E)/(N_E w_{EI}) + T_E \end{cases}$$

$$\nu_I(b) = \begin{cases} 0 & b < T_I(1-\lambda_E)/(N_E w_{EI}) + T_E \\ N_E w_{EI}\nu_E - T_I & b \ge T_I(1-\lambda_E)/(N_E w_{EI}) + T_E \end{cases},$$

where $\lambda_I = N_I N_E w_{IE} w_{EI}$, $\lambda_E = 2w_E + w_S$, and $\Lambda = 1 - \lambda_E + \lambda_I$. The conditions over b correspond to the boundary on which the inhibitory neurons become active. When $b \le T_E$, the input cannot overcome the leak term and the neurons do not fire.

The extension of this calculation to more than two nearest neighbors is straightforward and can be taken into account by scaling w_E with the number of nearest neighbors per neuron.

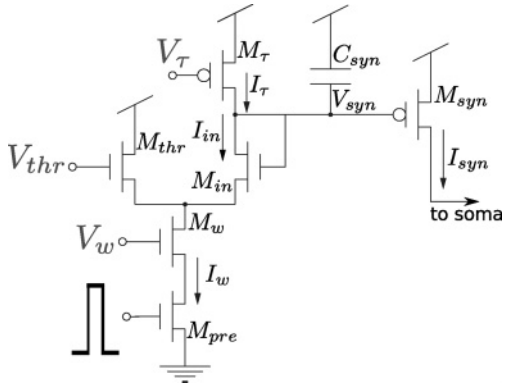


Figure 12: Schematics of the differential-pair integrator, often used for VLSI implementations of synapses. The nodes V_τ, V_{thr}, and V_w are the parameters (biases) of the synapse controlling, respectively, the time constant and the weight (both V_{thr} and V_w). The spikes arrive on the transistor M_{pre}, which acts as a digital switch, and the current flowing to the soma is regulated by V_{syn}.

Appendix B: The Differential-Pair-Integrator Synapse

The differential-pair integrator (DPI) is a VLSI implementation designed by Bartolozzi and Indiveri (2007) to mimic the dynamics of the synapse presented in equation 2.3. Its circuit is shown in Figure 12. The synapse is driven by incoming spikes arriving at the gate of the transistor M_{pls}, which acts as a digital switch. We assume that the spike is a box-shaped function $\Delta_{pls}(t)$ of duration t_{pls} and of unit height. By solving the circuit depicted in Figure 12 in the subthreshold regime, the equation governing the synaptic current becomes

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} I_{syn} + I_{syn} = \frac{I_\omega}{I_\tau} \frac{I_{syn}}{1 + \frac{I_{syn}}{I_{gain}}} \rho(t), \tag{B.1}$$

with the spike train $\rho(t) = \sum_k \Delta_{pls}(t - t_k)$. The currents I_τ and I_ω are the currents flowing through the transistor M_τ and M_w, respectively, and I_{gain} is the subthreshold current of a virtual p-type transistor of the same geometry as M_{syn} and whose gate voltage is set to V_{thr}. The time constant of the synapse is $\tau = \frac{C_{syn} U_T}{\kappa I_\tau}$, with U_T the thermal voltage, C_{syn} the capacitance of the synapse, and κ the subthreshold slope factor of the transistor. At steady state, the synaptic current is $\frac{I_{gain}}{I_\tau}(I_w - I_\tau)$. We see that if $I_\omega \gg I_\tau$, the synaptic current rises to values such that $I_{syn} \gg I_{gain}$. Applying this

condition to equation B.1 leads to the following linear differential equation for the synaptic current:

$$\tau \frac{\mathrm{d}}{\mathrm{d}t} I_{syn} + I_{syn} = \frac{I_\omega I_{gain}}{I_\tau} \rho(t). \tag{B.2}$$

Following similar arguments as for equation A.3 and by assuming that the pulse has a finite duration t_{pls}, one shows that

$$\langle I_{syn} \rangle \cong t_{pls} \frac{I_\omega I_{gain}}{I_\tau} \nu_{in}. \tag{B.3}$$

A quick comparison with equation A.3 reveals that $t_{pls} \frac{I_\omega I_{gain}}{I_\tau} = q_w$, where q_w was the scaling factor for the weight of the synapse. Since the differential-pair integrator synapse receives input as pulses on gate M_{pre}, the effective current flowing through the transistor M_w will be $I_w t_{pls} \nu_{in}$, and the condition for the linearity of equation B.1 becomes $I_\omega t_{pls} \nu_{in} \gg I_\tau$ with ν_{in} the mean frequency of the incoming spike trains. Therefore, the synaptic current I_{syn} as a function of ν_{in} is approximatively threshold linear. For most excitatory synapses with long time constants (more than 50 ms), this effect can be neglected. However, this nonlinearity is clearly observable in the inhibitory synapse due to shorter pulse durations t_{pls} resulting from our implementation. In this case, the effective synaptic current is given by

$$I_{syn}(t) \cong t_{pls} \frac{I_\omega I_{gain}}{I_\tau} \max\left(\nu_{in} - \frac{I_\tau}{I_\omega t_{pls}}, 0\right). \tag{B.4}$$

In practice, the threshold term is determined experimentally because of its dependence on the pulse duration, which cannot be measured experimentally.

Appendix C: Chip Settings Used During Parameter Translations

As a reference, the relevant biases fixed during the parameter translations are given below.

`nrf`= 0.25 V: Refractoriness bias of the soma, corresponding to a very short refractory period. (< 1 ms).
`nsf`= 0.75 V: Threshold V_{th} of the positive feedback circuit for the spike generation, corresponding to an effective threshold of $\Theta = 1.1$ V.
`nplsloc`= 0.40 V: Local excitatory synapse pulse extender bias, roughly corresponding to pulses in the μs range.
`nplslocinh`= 0.40 V: Local inhibitory synapse pulse extender bias.

`psynlocinhw`= 2.50 V: Weight bias of the inhibitory differential-pair integrator synapse. Because the weight bias V_w of the differential-pair integrator synapse also affects the nonlinearity described above, we have used the threshold bias V_{thr} of the differential-pair integrator synapse (controlling I_{gain}) to set weight of the inhibitory synapse.
`nsynlocth`= 3.00 V: Threshold bias V_{thr} of the local excitatory differential-pair integrator synapses.
`nsynaerpls`= 0.40 V: address event representation excitatory synapse pulse extender bias.
`nsynstdth`= 2.85 V: Threshold bias V_{thr} of the excitatory address event representation differential-pair integrator synapse.

Furthermore, the adaptation circuit in the soma and the short-term depression circuit in the differential-pair integrator synapse were turned off (see Indiveri et al., 2006). A full list of biases, as well as the list of values measured from layout ($\frac{W}{L}$ and capacitances) can be obtained from the authors on request.

Acknowledgments

This work was supported by the DAISY (FP6-2005-015803) EU grant, the Swiss National Science Foundation (PMPD2-110298/1), and the EU ICT Grant ICT-231168-SCANDLE Acoustic SCene ANalysis for Detecting Living Entities. We thank D. Fasnacht for the design of the AER monitor/sequencer board and the reviewers for their useful comments.

References

Abarbanel, H., Creveling, D., Farsian, R., & Kostuk, M. (2009). Dynamical state and parameter estimation. *SIAM Journal on Applied Dynamical Systems, 8*, 1341–1381.

Ahmed, B., Anderson, J., Douglas, R., Martin, K., & Whitteridge, D. (1998). Estimates of the net excitatory currents evoked by visual stimulation of identified neurons in cat visual cortex. *Cerebral Cortex, 8*, 462–476.

Alvado, L., Tomas, J., Saighi, S., Renaud-Le Masson, S., Bal, T., Destexhe, A., et al. (2004). Hardware computation of conductance-based neuron models. *Neurocomputing, 58–60*, 109–115.

Amari, S., & Arbib, M. A. (1977). Competition and cooperation in neural nets. In J. Metzler (Ed.), *Systems Neuroscience* (pp. 119–165). Orlando, FL: Academic Press.

Arthur, J., & Boahen, K. (2004, July). Recurrently connected silicon neurons with active dendrites for one-shot learning. In *IEEE International Joint Conference on Neural Networks* (Vol. 3, pp. 1699–1704). Piscataway, NJ: IEEE.

Arthur, J., & Boahen, K. (2007). Synchrony in Silicon: The Gamma Rhythm. *IEEE Transactions on Neural Networks, 18* 1815–1825.

Bartolozzi, C., & Indiveri, G. (2006). Selective attention implemented with dynamic synapses and integrate-and-fire neurons. *NeuroComputing, 69*(16–18), 1971–1976.

Bartolozzi, C., & Indiveri, G. (2007). Synaptic dynamics in analog VLSI. *Neural Computation, 19*(10), 2581–2603.

Bartolozzi, C., Mitra, S., & Indiveri, G. (2006). An ultra low power current–mode filter for neuromorphic systems and biomedical signal processing. In *Biomedical Circuits and Systems Conference, BIOCAS 2006* (pp. 130–133). Amsterdam: Elsevier.

Ben-Yishai, R., Lev Bar-Or, R., & Sompolinsky, H. (1995). Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences of the USA, 92*(9), 3844–3848.

Bower, J., Beeman, D., & Wylde, A. (1998). *The book of GENESIS: Exploring realistic neural models with the GEneral NEural SImulation System*. New York: Springer.

Brillinger, D. (1988). Maximum likelihood analysis of spike trains of interacting nerve cells. *Biological Cybernetics, 59*3, 189–200.

Brüderle, D. (2009). *Neuroscientific Modeling with a Mixed-Signal VLSI Hardware System*. Unpublished doctoral dissertation, University of Heidelberg.

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience, 8*(3), 183–208.

Camilleri, P., Giulioni, M., Dante, V., Badoni, D., Indiveri, G., Michaelis, B., et al. (2007). A neuromorphic aVLSI network chip with configurable plastic synapses. In *Proceedings of the 7th International Conference on Hybrid Intelligent Systems* (pp. 296–301). Los Alamitos, CA: IEEE Computer Society.

Chicca, E., Whatley, A. M., Dante, V., Lichtsteiner, P., Delbrück, T., Del Giudice, P., et al. (2007). A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity. *IEEE Transactions on Circuits and Systems I, 5*(54), 981–993.

Choi, T.Y.W., Merolla, P. A., Arthur, J. V., Boahen, K. A., & Shi, B. E. (2005). Neuromorphic implementation of orientation hypercolumns. *IEEE Transactions on Circuits and Systems I, 52*(6), 1049–1060.

Culurciello, E., Etienne-Cummings, R., & Boahen, K. (2003). A biomorphic digital image sensor. *IEEE Journal of Solid State Circuits, Part 2, 38*, 281–294.

Davison, A., Brüderle, D., Eppler, J., Kremkow, J., Muller, E., Pecevski, D., et al. (2008). PyNN: A common interface for neuronal network simulators. *Front. Neuroinform., 2*, 11.

Dayan, P., & Abbott, L. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press.

Destexhe, A., Mainen, Z., & Sejnowski, T. (1998). Methods in Neuronal Modelling, from ions to networks. In C. Koch & I. Segev (Eds.), *Methods in neuronal modeling* (pp. 1–25). Cambridge, MA: MIT Press.

Douglas, R., & Mahowald, M. (1995). Silicon neurons. In M. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 282–289). Cambridge, MA: MIT Press.

Douglas, R., Mahowald, M., & Martin, K. (1994). Silicon neurons. In M. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 282–289). Cambridge, MA: MIT Press.

Douglas, R., & Martin, K. (2004). Neural circuits of the neocortex. *Annual Review of Neuroscience, 27*, 419–451.

Ermentrout, B. (1994). Reduction of conductance-based models with slow synapses to neural nets. *Neural Comput., 6*(4), 679–695.

Farquhar, E., & Hasler, P. (2005). A bio-physically inspired silicon neuron. *IEEE Transactions on Circuits and Systems, 52*(3), 477–488.

Fasnacht, D., Whatley, A., & Indiveri, G. (2008, May). A serial communication infrastructure for multi-chip address event system. In *International Symposium on Circuits and Systems, ISCAS 2008* (pp. 648–651). Piscataway, NJ: IEEE.

Folowosele, F., Etienne-Cummings, R., & Hamilton, T. (2009, Nov.). A CMOS switched capacitor implementation of the Mihalas-Niebur neuron. In *Biomedical Circuits and Systems Conference* (pp. 105–108). Piscataway, NJ: IEEE.

Fourcaud, N., & Brunel, N. (2002). Dynamics of the firing probability of noisy integrate-and-fire neurons. *Neural Computation, 14*(9), 2057–2110.

Fusi, S., & Mattia, M. (1999). Collective behavior of networks with linear (VLSI) integrate and fire neurons. *Neural Computation, 11*, 633–652.

Hahnloser, R., Sarpeshkar, R., Mahowald, M., Douglas, R., & Seung, S. (2000). Digital selection and analog amplification co-exist in an electronic circuit inspired by neocortex. *Nature, 405*(6789), 947–951.

Hahnloser, R.H.R., Seung, H. S., & Slotine, J. J. (2000). Permitted and forbidden sets in symmetric threshold-linear networks. *Neural Computation, 15*, 621–638.

Hansel, D., & Sompolinsky, H. (1998). Modeling feature selectivity in local cortical circuits. In C. Koch & I. Segev (Eds.), *Methods in neuronal modeling*. Cambridge, MA: MIT Press.

Hines, M., & Carnevale, N. (1997). The NEURON simulation environment. *Neural Computation, 9*(6), 1179–1209.

Huys, Q., Ahrens, M., & Paninski, L. (2006). Efficient estimation of detailed single-neuron models. *Journal of Neurophysiology, 96*(2), 872–980.

Hynna, K., & Boahen, K. (2001). Space–rate coding in an adaptive silicon neuron. *Neural Networks, 14*, 645–656.

Hynna, K. M., & Boahen, K. (2006, May). Neuronal ion-channel dynamics in silicon. In *International Symposium on Circuits and Systems* (pp. 3614–3617). Amsterdam: Elsevier.

Indiveri, G. (2003, May). A low-power adaptive integrate-and-fire neuron circuit. In *International Symposium on Circuits and Systems* (pp. IV-820–IV-823). Piscataway, NJ: IEEE.

Indiveri, G., Chicca, E., & Douglas, R. (2006). A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Transactions on Neural Networks, 17*(1), 211–221.

Keat, J., Reinagel, P., Reid, R., & Meister, M. (2001). Predicting every spike: A model for the responses of visual neurons. *Neuron, 30*, 803–817.

Lazzaro, J., Wawrzynek, J., Mahowald, M., Sivilotti, M., & Gillespie, D. (1993). Silicon auditory processors as computer peripherals. *IEEE Transactions on Neural Networks, 4*, 523–528.

Lichtsteiner, P., Posch, C., & Delbruck, T. (2008). An 128 $\times$ 128 120dB 15μs-latency temporal contrast vision sensor. *IEEE J. Solid State Circuits, 43*(2), 566–576.

Livi, P., & Indiveri, G. (2009, May). A current-mode conductance-based silicon neuron for Address-Event neuromorphic systems. In *International Symposium on Circuits and Systems* (pp. 2898–2901). Piscataway, NJ: IEEE.

Lyon, R., & Mead, C. (1988). An analog electronic cochlea. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 36*(7), 1119–1134.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation, 14*(11), 2531–2560.

Mahowald, M., & Douglas, R. (1991). A silicon neuron. *Nature, 354*, 515–518.

Mahowald, M., & Mead, C. (1989). Silicon retina. In C. Koch (Ed.), *Analog (VLSI) and neural systems* (pp. 257–278). Reading, MA Addison-Wesley.

Massoud, T., & Horiuchi, T. (2009). A neuromorphic head direction cell system. In *International Symposium on Circuits and Systems* (pp. 565–568). Piscataway, NJ: IEEE.

Mead, C. (1989). *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley.

Mead, C., & Mahowald, M. (1988). A silicon model of early visual processing. *Neural Networks, 1*, 91–97.

Mitra, S., Fusi, S., & Indiveri, G. (2006, May). A VLSI spike-driven dynamic synapse which learns only when necessary. In *Proceedings of the IEEE International Symposium on Circuits and Systems* (pp. 2777–2780). Piscataway, NJ: IEEE.

Neftci, E., Chicca, E., Indiveri, G., Cook, M., & Douglas, R. (2010). State-dependent sensory processing in networks of VLSI spiking neurons. In *International Symposium on Circuits and Systems* (pp. 2789–2792). Piscataway, NJ: IEEE.

Neftci, E., Chicca, E., Indiveri, G., Slotine, J.-J., & Douglas, R. (2008). Contraction properties of VLSI cooperative competitive neural networks of spiking neurons. In J. Platt, D. Koller, Y. Singer & S. Roweis (Eds.), *Advances in neural information processing systems, 20* (pp. 1073–1080). Cambridge, MA: MIT Press.

Neftci, E., & Indiveri, G. (2010). A device mismatch reduction method for VLSI spiking neural networks. In *Biomedical Circuits and Systems Conference*. Piscataway, NJ: IEEE.

Okatan, M., Wilson, M. A., & Brown, E. N. (2005). Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Computation, 17*(9), 1927–1961.

Paninski, L., Pillow, J. W., & Simoncelli, E. P. (2004). Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Comput., 16*(12), 2533–2561.

Pavasović, A., Andreou, A., & Westgate, C. (1994) Characterization of subthreshold MOS mismatch in transistors for VLSI systems0. *Journal of VLSI Signal Processing, 8*(1), 75–85.

Pecevski, D., Natschläger, T., & Schuch, K. (2008). PCSIM: A parallel simulation environment for neural circuits fully integrated with Python. *Frontiers Neuroinf.*, 3.

Posch, C., Matolin, D., & Wohlgenannt, R. (2010). A QVGA 143dB dynamic range asynchronous address-event PWM dynamic image sensor with lossless pixel-level video compression. In *2010 IEEE ISSCC Digest of Technical Papers* (pp. 400–401). Piscataway, NJ: IEEE.

Rasche, C., & Douglas, R. (2000). An improved silicon neuron. *Analog Integrated Circuits and Signal Processing, 23*(3), 227–236.

Rastogi, M., Garg, V., & Harris, J. (2009, May). Low power integrate and fire circuit for data conversion. In *International Symposium on Circuits and Systems* (pp. 2669–2672). Piscataway, NJ: IEEE.

Renart, A., Brunel, N., & Wang, X. (2003). Mean field theory of irregularly spiking neuronal populations and working memory in recurrent cortical networks. In J. Feng (Ed.), *Computational neuroscience: A comprehensive approach* (pp. 431–490). Boca Raton, FL: Chapman and Hall.

Renaud, S., Tomas, J., Bornat, Y., Daouzli, A., & Saïghi, S. (2007). Neuromimetic ICs with analog cores: An alternative for simulating spiking neural networks. In *International Symposium on Circuits and Systems* (pp. 3355–3358). Piscataway, NJ: IEEE.

Russell, A., Orchard, G., & Etienne-Cummings, R. (2007). Configuring of spiking central pattern generator networks for bipedal walking using genetic algorithms. In *International Symposium on Circuits and Systems* (pp. 1525–1528).

Rutishauser, U., & Douglas, R. (2009). State-dependent computation using coupled recurrent networks. *Neural Computation, 21*, 478–509.

Schemmel, J., Fieres, J., & Meier, K. (2008). Wafer-scale integration of analog neural networks. In *Proceedings of the IEEE International Joint Conference on Neural Networks.* Piscataway, NJ: IEEE.

Schemmel, J., Meier, K., & Mueller, E. (2004, July). A new VLSI model of neural microcircuits including spike time dependent plasticity. In *Proceedings of the IEEE International Joint Conference on Neural Networks* (Vol. 3, pp. 1711–1716). Piscataway, NJ: IEEE.

Schwartz, E. (1993). *Computational neuroscience*. Cambridge, MA: MIT Press.

Serrano-Gotarredona, R., Oster, M., Lichtsteiner, P., Linares-Barranco, A., Paz-Vicente, R., Gómez-Rodríguez, F., et al. (2009, September). CAVIAR: A 45k neuron, 5m synapse, 12g connects/s AER hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking. *IEEE Transactions on Neural Networks, 20*(9), 1417–1438.

Shriki, O., Hansel, D., & Sompolinsky, H. (2003). Rate models for conductance-based cortical neuronal networks. *Neural Comput., 15*(8), 1809–1841.

Silver, R., Boahen, K., Grillner, S., Kopell, N., & Olsen, K. (2007). Neurotech for neuroscience: Unifying concepts, organizing principles, and emerging tools. *Journal of Neuroscience, 27*(44), 11807–11819.

Simoni, M., Cymbalyuk, G., Sorensen, M., Calabrese, R., & DeWeerth, S. (2004). A multiconductance silicon neuron with biologically matched dynamics. *IEEE Transactions on Biomedical Engineering, 51*(2), 342–354.

Steimer, A., Maass, W., & Douglas, R. (2009). Belief propagation in networks of spiking neurons. *Neural Computation, 21*, 2502–2523.

Tsividis, Y. (1998). *Operation and modeling of the MOS transistor*. New York: McGraw-Hill.

Tuckwell, H. (1988). *Introduction to theoretical neurobiology*. Cambridge: Cambridge University Press.

van Schaik, A. (2001). Building blocks for electronic spiking neural networks. *Neural Networks, 14*(6–7) 617–628.

van Schaik, A., & Liu, S.-C. (2005, May). AER EAR: A matched silicon cochlea pair with address event representation interface. In *International Symposium on Circuits and Systems* (Vol. 5, pp. 4213–4216). Piscataway, NJ: IEEE.

Wijekoon, J., & Dudek, P. (2008). Compact silicon neuron circuit with spiking and bursting behaviour. *Neural Networks, 21*(2–3), 524–534.

Yu, T., & Cauwenberghs, G. (2009). Analog VLSI neuromorphic network with programmable membrane channel kinetics. In *International Symposium on Circuits and Systems, ISCAS 2009* (pp. 349–352). Piscataway, NJ: IEEE.

Yuille, A. L., & Grzywacz, N. M. (1989). A winner-take-all mechanism based on presynaptic inhibition feedback. *Neural Comput., 1*(3), 334–347.

Received August 25, 2010; accepted March 3, 2011.